

11-13-00  
A  
Customer No. 20350  
TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, 8<sup>th</sup> Floor  
San Francisco, California 94111-3834  
(650) 326-2400

ASSISTANT COMMISSIONER FOR PATENTS  
BOX PATENT APPLICATION  
Washington, D.C. 20231



Attorney Docket No. 020508-000100US

"Express Mail" Label No. EL381322205US

Date of Deposit: November 9, 2000

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is the

- ☒ patent application of  
☐ continuation patent application of  
☐ divisional patent application of  
☐ continuation-in-part patent application of

Inventor(s)/Applicant Identifier: Gregg Williams, Mace Wolf, and Jim Tucker

For: SUBSCRIPTION MEMBERSHIP MARKETING APPLICATION FOR THE INTERNET

☐ This application claims priority from each of the following Application Nos./filing dates:

the disclosure(s) of which is (are) incorporated by reference.

☐ Please amend this application by adding the following before the first sentence: "This application is a ☐ continuation ☐ continuation-in-part of and claims the benefit of U.S. Provisional Application No. 60/\_\_\_\_\_, filed \_\_\_\_\_, the disclosure of which is incorporated by reference."

Enclosed are:

- ☒ 22 page(s) of specification  
☒ 1 page of Title Sheet  
☒ 6 page(s) of claims  
☒ 1 page of Abstract  
☒ 12 sheet(s) of ☐ formal ☒ informal drawing(s).  
☒ Appendix, A, B, C, D, and E (included, binder).

	(Col. 1)	(Col. 2)
FOR:	NO. FILED	NO. EXTRA
BASIC FEE		
TOTAL CLAIMS	25 - 20	= *5
INDEP. CLAIMS	5 - 3	= *0
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENTED		

\* If the difference in Col. 1 is less than 0, enter "0" in Col. 2.

SMALL ENTITY

RATE	FEE
	\$355.00
x \$9.00 =	\$45.00
x \$40.00 =	\$0.00
+ \$135.00 =	
TOTAL	\$400.00

OTHER THAN  
SMALL ENTITY

RATE	FEE
	\$710.00
x \$18.00 =	
x \$80.00 =	
+ \$270.00 =	
TOTAL	

Please charge Deposit Account No. 20-1430 as follows:

- ☒ Filing fee \$400.00  
☒ Any additional fees associated with this paper or during the pendency of this application.  
☐ The issue fee set in 37 CFR 1.18 at or before mailing of the Notice of Allowance, pursuant to 37 CFR 1.311(b)

☐ A check for \$\_\_\_\_\_ is enclosed.  
2 extra copies of this sheet are enclosed.

Telephone:  
(650) 326-2400

Facsimile:  
(650) 326-2422

Respectfully submitted,  
TOWNSEND and TOWNSEND and CREW LLP

Paul A. Durdik  
Reg No.: 37,819  
Attorneys for Applicants

**PATENT APPLICATION**

**SUBSCRIPTION MEMBERSHIP MARKETING APPLICATION FOR  
THE INTERNET**

Inventor(s):

Gregg Williams  
4406 20<sup>th</sup> Street, San Francisco, CA 94114  
US Citizen

Mace Wolf  
1612 Allston Way, Berkeley, CA 94703  
US Citizen

Jim Tucker  
304 Poplar, Mill Valley, CA 94941  
US Citizen

Assignee:

Integrated Marketing Technology  
50 Osgood Place Suite 330  
San Francisco, California 94133

A California corporation

Entity: Small



## 5

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the xerographic reproduction by anyone of the patent document or the patent disclosure in exactly the form it appears in the U.S. Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

The following appendicies are being filed with this application, the entire contents of which are herein incorporated by reference for all purposes:

15

Appendix C (200 pages) - ASP Source Code;

Appendix E (58 pages) - Users Manual.

## 20

The present invention relates generally to network based marketing techniques, and specifically to techniques for managing and analyzing subscriber information in a networked computer environment.

25

Few could foresee the rapid expansion of direct marketing campaigns just a few years ago. Direct marketers use a variety of marketing techniques to provide information about new products and services to consumers. Direct mail, internet advertising, television, radio, newspaper, and magazine advertising provide channels of communication for product information to consumers. Surveys, purchasing information, transaction information and others provide information about consumer behaviour to the direct marketer. Computers and televisions that now have a place in our homes and offices enable product manufacturers, wholesalers and retailers to “take their case directly to the public,” by providing a communications medium with direct access to the consumer. Never before have consumers

been offered such a wide variety of choices in the products for their homes, work and recreation.

As computer automation continues to provide an ever increasing access to consumers, marketers have discovered that conventional marketing information methods do not readily scale to the exponential proliferation of information accompanying direct marketing campaigns. For example, results from multiple targeted marketing campaigns must be sorted, assimilated, analyzed and managed on a continuing basis.

Conventional techniques for information management exhibit inefficiencies when applied to the subscription marketing campaign field. For example, conventional techniques are frequently embodied in specialized software, or require custom solutions. Many conventional approaches require the user to learn and use ad hoc queries.

What is really needed are subscription information management techniques designed to meet the needs of subscription marketers.

## SUMMARY OF THE INVENTION

The present invention provides network based reporting techniques designed to meet the needs of subscription based marketers. Clients access, edit and report on their subscriber database using the Internet, for example. In a representative embodiment, a plurality of client computers connect to the Internet using a local area network (LAN), for example. The client computers pass security protocols before gaining access to web, application and database servers. Once connected, the client computers are able to retrieve and display information using a web browser. Database contents are denormalized to provide the capability to generate reports based upon relatively large amounts of information.

In a representative embodiment, the present invention provides a method for analyzing subscription information. The method comprises receiving a plurality of subscription information and archiving the plurality of subscription information into a repository. Retrieving from the repository a plurality of selected subscription information and transforming the plurality of selected subscription information into transition table format information are also part of the method. Further, the method includes populating a facts database with the transition format information to produce one or more facts tables with a summarization of the transition format information. Denormalizing the summarization of the transition format information to produce a plurality of denormalized information is also part

of the method. The method also includes providing a report based upon the denormalized information.

In a specific embodiment, the denormalization comprises populating selected information from one or more facts tables to a key table. The facts tables are stored in a facts table database, for example. The key table is stored in a grouping tables database. The plurality of facts tables are one of a source category table, a source table, a campaign table, and a panels table.

In a specific embodiment, the method further comprises receiving input of a criteria for one or more reports and providing the report based upon the denormalized information according to the criteria. In another specific embodiment, the method further comprises receiving input of a criteria for one or more tables and configuring the one or more tables based upon the criteria.

In another specific embodiment, the method further comprises receiving input of a username and a password and verifying whether the input of the username and a password corresponds to a user authorized to access reporting procedures. If the user is determined to be authorized, access to reporting procedures is provided to the user. In a specific embodiment, the method further comprises verifying whether the input of the username and a password corresponds to a user authorized to access administrative procedures. If the user is determined to be authorized access to administrative procedures is provided to the user.

Alternative embodiments include an apparatus having a plurality of means for performing the above described method.

In another representative embodiment, the present invention provides a computer program product for analyzing subscription information. The computer program product comprises a variety of software program code. For example, code that receives a plurality of subscription information and code that archives the plurality of subscription information into a repository are included in the program product. Code that retrieves from the repository a plurality of selected subscription information and code that transforms the plurality of selected subscription information into transition table format information are also part of the program product. The program product also includes code that populates a facts database with the transition format information to produce one or more facts tables with a summarization of the transition format information and code that denormalizes the summarization of the transition format information to produce a plurality of denormalized information, the denormalization comprising populating selected information from the facts

tables, stored in a facts table database, to a key table, stored in a grouping tables database. Further, code that provides a report based upon the denormalized information is also included in the program product. A computer readable storage medium for holding the codes also comprises the program product.

5 In a yet further representative embodiment, the present invention provides a system for analyzing subscription information. The system comprises a variety of components. For example, one or more clients are connected to a web server by a computer network. An application server and a database server are connected to the web server by a communication medium, such as a computer network, for example. The clients receive  
10 subscription information and forward the subscription information via the web server and the application server to the database server, which archives subscription information into a repository. The application server retrieves selected subscription information from the repository and transforms the selected subscription information into transition table format information. The database server populates a facts database with the transition format  
15 information in order to produce one or more facts tables with a summarization of the transition format information. The application server denormalizes the summarization of the transition format information in order to produce denormalized information. This denormalization comprises populating selected information from one or more facts tables, which can be stored in a facts table database, into a key table, which is stored in a grouping  
20 tables database, for example. The application server provides a report based upon the denormalized information.

Numerous benefits are achieved by way of the present invention over conventional techniques. The present invention can provide network based reporting techniques designed to meet the needs of subscription marketers. Embodiments according to  
25 the present invention can provide real-time marketing information analysis techniques, which can provide information consumers within a business enterprise with up to the moment information and analysis for mission critical applications.

In specific embodiments according to the present invention, denormalization and report optimizing processing provides improved reporting capabilities by reducing at  
30 least one of a number of joins of database tables that must be made and a number of records that must be reported. These reductions enable specific embodiments to provide reports more quickly than conventional techniques. Specific embodiments deliver reports over computer networks, such as the Internet. Further, specific embodiments can produce a report from a relatively large number of transactions within a reasonable delay time.

These and other benefits are described throughout the present specification. A further understanding of the nature and advantages of the invention herein may be realized by reference to the remaining portions of the specification and the attached drawings.

5

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 depicts a simplified block diagram of a distributed computer network which may incorporate an embodiment of the present invention;

Fig. 2 depicts a simplified block diagram of a computer system which may incorporate an embodiment of the present invention;

10 Fig. 3 illustrates a diagram of representative processing, manipulating, storing and loading of data in a specific embodiment according to the present invention;

Fig. 4 illustrates a diagram of representative data acquisition, processing and loading in a specific embodiment according to the present invention;

15 Figs. 5A-5B illustrate diagrams of a representative online administrative system in a specific embodiment according to the present invention;

Fig. 6 illustrates a diagram of a representative administrative framework that provides a grouping hierarchy in a specific embodiment according to the present invention;

Fig. 7 illustrates a diagram of representative reporting process in a specific embodiment;

20 Fig. 8 illustrates a diagram showing a representative hyperlink chain linking reports in a specific embodiment according to the present invention;

Fig. 9 illustrates a diagram of a representative access security for a database and reporting system in a specific embodiment according to the present invention;

25 Fig. 10 illustrates a diagram of a representative process for denormalizing database tables in a specific embodiment according to the present invention; and

Fig. 11 illustrates a diagram of a representative process for optimizing report generation in a specific embodiment according to the present invention.

## DESCRIPTION OF THE SPECIFIC EMBODIMENTS

30 The present invention provides network based reporting techniques designed to meet the needs of subscription based marketers. Clients access, edit and report on their subscriber database using the Internet, for example. In a representative embodiment, a plurality of client computers connect to the Internet using a local area network (LAN), for

example. The client computers pass security protocols before gaining access to web, application and database servers. Once connected, the client computers are able to retrieve and display information using a web browser. Database contents are denormalized to provide the capability to generate reports based upon relatively large amounts of information.

5                Techniques are discussed for managing the process of information gathering, analysis and storage. Fig. 1 is a simplified block diagram of a distributed computer network 10 which may incorporate an embodiment of the present invention. Computer network 10 includes a number of computer systems 12, 15, 17, 14-1, 14-2, and 14-3. Computer systems 12, 14-1, 14-2, and 14-3 are coupled to a communication network 16 via a plurality of  
10        communication links 18. The computer systems include a plurality of client computer systems 14-1, 14-2, and 14-3, and a plurality of server computer systems 12, 15, and 17. However, in particular embodiments, two or more of server systems 12, 15, and 17 may co-reside on a single computer. Client systems 14-1, 14-2, or 14-3 typically request information  
15        from a server computer system, which performs processing in response to the client request and provides the requested information to the client systems. For this reason, servers typically have more computing and storage capacity than client systems. However, a particular computer system may act as both as a client or a server depending on whether the computer system is requesting or providing information.

              Communication network 16 provides a mechanism for allowing the various  
20        components of distributed network 10 to communicate and exchange information with each other. Communication network 16 may itself be comprised of many interconnected computer systems and communication links. Communication links 18 may be hardwire links, optical links, satellite or other wireless communications links, wave propagation links, or any other mechanisms for communication of information. While in one embodiment, communication  
25        network 16 is the Internet, in other embodiments, communication network 16 may be any suitable computer network. An optional firewall (not shown) having a port for connecting thereto may be provided between the server computer 12 and communications network 16. Distributed computer network 10 depicted in Fig. 1 is merely illustrative of an embodiment incorporating the present invention and does not limit the scope of the invention as recited in  
30        the claims. One of ordinary skill in the art would recognize other variations, modifications, and alternatives. For example, more than one server system 12 may be coupled to communication network 16.

              Computer server systems 12, 15, and 17 may be configured to perform a plurality of functions according to the teachings of the present invention. These functions are

typically performed by software code modules executing on the computer server systems 12, 15, and 17. The functions may also be performed by hardware modules coupled to server systems 12, 15, and 17, or by a combination of software and hardware modules. Functions performed by computer server systems 12, 15 and 17 include storing and retrieving  
5 information in database 109, and performing administrative functions, report generation functions, and other functions according to the teachings of the present invention. Details related to the various functions performed by server systems 12, 15 and 17 are described below.

Client systems 14-1, 14-2, or 14-3 are used to interact with one or more of  
10 server systems 12, 15 and 17 using a program called a browser. The browser may be one of Internet Explorer™, by Microsoft™, or Navigator™ by Netscape™, or the like, but in a presently preferred embodiment, the browser comprises support for HTML 2.0 or higher, Active X and/or Java functions. For example, client systems 14-1, 14-2, or 14-3 may be used by users to access administrative functions and features online. Client systems 14-1, 14-2, or  
15 14-3 may also be used by users to obtain reports using online statistical report capabilities of specific embodiments. Once a user has provided a unique user identification and password, the user may use client system 14-1, 14-2, or 14-3 to participate in administrative and report generating capabilities of server systems 12, 15 and 17. Details related to the various functions performed by client systems 14-1, 14-2, or 14-3 are described below.

20 According to the teachings of the present invention, computer system 12 is a web server system responsible for providing web pages to users at clients 14-1, 14-2 and 14-3, and storing information received from the clients in a format which allows later query or retrieval of the information. In a presently preferred embodiment, web server system 12, comprises Microsoft™ Internet Information Services (IIS) and Seagate™ Crystal Reports.  
25 The term "server system" as used in this application may refer to a single server system as depicted in Fig. 1, or may refer to one or more server systems distributed within computer network 10. If the functions are distributed across a plurality of computers, each comprises a copy of Microsoft™ Internet Information Services (IIS). Accordingly, functions or tasks performed by the present invention may be distributed to one or more servers coupled to  
30 communication network 16. According to a specific embodiment, the servers may be isolated behind firewalls for security purposes and communication between the servers may be encoded and encrypted.

According to the teachings of the present invention, computer server system 15 is an application server system responsible for providing administrative and reporting

capabilities. Application server system 15 comprises one or more administrative processes and one or more reporting processes. Administrative processes provide users with capabilities to create and edit database tables that contain marketing information and the like, provide security and access control functions, track usage, and the like. Reporting processes provide users with the capabilities to generate, and customize reports that provide insight into marketing information and the like. The administrative processes and reporting processes may be implemented in a variety of different ways, but in a presently preferred embodiment are implemented using Microsoft<sup>TM</sup> Visual Basic (VB) and PERL script programs (scripts).

According to the teachings of the present invention, computer system 17 is a database server system responsible for providing database support. In a specific embodiment, the transaction information may be stored in a database 109 coupled to database server system 17. The term “database 109” as used in this application may refer to a single information store or to a plurality of information stores distributed within computer network 10. For example, database 109 be locally coupled to database server system 17 or may be distributed across computer network 10 and accessed by database server system 17 via communication network 16. The database 109 may be a relational database, an object-relational database, an object-oriented database, a knowledge base, a text file, or any other way to store information. However, in a presently preferred embodiment, the database server system 17 comprises Oracle<sup>TM</sup> database and compatible hardware. In a specific embodiment of the present invention, database 109 is a relational database configured to store information according to a database schema. The information stored in database 109 may also be used to keep track of the various steps of the information analysis and storage process. For example, the status or progress of any particular step of the information acquisition process can be ascertained from the information stored in database 109. It should be apparent that although Fig. 1 depicts database 109 as a single entity, in a specific embodiment of the present invention, database 109 may comprise a plurality of databases.

Fig. 2 is a simplified block diagram of an exemplary computer system 20 capable of embodying one or more of client computer systems 14-1, 14-2, and 14-3, and server computer systems 12, 15 and 17 according to the present invention. Computer system 20 typically includes at least one processor 24, which communicates with a number of peripheral devices via bus subsystem 22. These peripheral devices typically include a storage subsystem 32, comprising a memory subsystem 34 and a file storage subsystem 40, user interface input devices 30, user interface output devices 28, and a network interface subsystem 26. The input and output devices allow user interaction with computer system 20.



It should be apparent that the user may be a human user, a device, another computer, and the like. Network interface subsystem 26 provides an interface to outside networks, including an interface to communication network 16, and is coupled via communication network 16 to corresponding interface devices in other computer systems.

5           User interface input devices 30 may include a keyboard, pointing devices such as a mouse, trackball, touchpad, or graphics tablet, a scanner, a barcode scanner for scanning article barcodes, a touchscreen incorporated into the display, audio input devices such as voice recognition systems, microphones, and other types of input devices. In general, use of the term “input device” is intended to include all possible types of devices and ways to input  
10 information into computer system 20 or onto computer network 16.

          User interface output devices 28 may include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices. The display subsystem may be a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), or a projection device. The display subsystem may also provide non-visual display such as  
15 via audio output devices. In general, use of the term “output device” is intended to include all possible types of devices and ways to output information from computer system 20 to a human or to another machine or computer system.

          Storage subsystem 32 stores the basic programming and data constructs that provide the functionality of the various systems embodying the present invention. For  
20 example, the various modules implementing the functionality of the present invention may be stored in storage subsystem 32. These software modules are generally executed by processor(s) 24. In a distributed environment, the software modules may be stored on a plurality of computer systems and executed by processors of the plurality of computer systems. Storage subsystem 32 also provides a repository for storing the various databases  
25 storing information according to the present invention. Storage subsystem 32 typically comprises memory subsystem 34 and file storage subsystem 40.

          Memory subsystem 34 typically includes a number of memories including a main random access memory (RAM) 38 for storage of instructions and data during program execution and a read only memory (ROM) 36 in which fixed instructions are stored. File  
30 storage subsystem 40 provides persistent (non-volatile) storage for program and data files, and may include a hard disk drive, a floppy disk drive along with associated removable media, a Compact Digital Read Only Memory (CD-ROM) drive, an optical drive, removable media cartridges, and other like storage media. One or more of the drives may be located at remote locations on other connected computers at another site on communication network 16.

Information stored according to the teachings of the present invention may also be stored by file storage subsystem 40.

Bus subsystem 22 provides a mechanism for letting the various components and subsystems of computer system 20 communicate with each other as intended. The various subsystems and components of computer system 20 need not be at the same physical location but may be distributed at various locations within distributed network 10. Although bus subsystem 22 is shown schematically as a single bus, alternate embodiments of the bus subsystem may utilize multiple busses.

Computer system 20 itself can be of varying types including a personal computer, a portable computer, a workstation, a computer terminal, a network computer, a television, a mainframe, or any other data processing system. Due to the ever-changing nature of computers and networks, the description of computer system 20 depicted in Fig. 2 is intended only as a specific example for purposes of illustrating the preferred embodiment of the present invention. Many other configurations of a computer system are possible having more or less components than the computer system depicted in Fig. 2. Client computer systems 14 and one or more of server computer systems 12, 15 and 17 generally have the same configuration as shown in Fig. 2, with the server systems generally having more storage capacity and computing power than the client systems.

Fig. 3 illustrates a diagram of representative processing, manipulating, storing and loading of data in a specific embodiment according to the present invention. Fig. 3 illustrates a plurality of input data 101, comprising fulfillment data 101a, production data 101b, and client data 101c. Fulfillment data 101a is produced by a business process used to process an order or transaction resulting from a subscription campaign, for example. The fulfillment data 101a can be created by an "in-house" system operated by a subscription-based business or an outside service provider. Fulfillment data includes transaction and summary level statistical and coded information related to subscription orders. Fulfillment data 101a may be in a variety of formats in specific embodiments, but is preferably alpha/numeric. Examples of data feed elements include sourcekey identifiers, subscription order quantities, dollar values, demographics, and the like.

Production data 101b is produced by a business process (complimentary to fulfillment) having a function to physically manage an order shipment process and capture information related to shipment transactions. This type of process may be managed by a subscription-based business, or an outside service provider. Production data 101b may be in a variety of formats in specific embodiments, but is preferably alpha/numeric. Examples of

production data include product shipped, shipment date, location, subscription term, and the like.

Client data 101c comprises data related to a subscription transaction or data that is supplied by a customer. Client data 101c is supplied either directly from a subscription-based business or via a service provider. Examples include demographics, web-hit information, address information, list information, and the like.

Fig. 3 illustrates a data loading process 102 comprising of the archiving of low-level input files and the processing of these files. Data loading process 102 processes input files 101 to produce one or more transition tables 103. In a specific embodiment, the data loading application is implemented using the PERL programming language, data transformation service (DTS) and structured query language (SQL) loader script programs. During data loading process 102, input data 101 in import files is identified, validated and formatted. The transition tables 103 store input data 101 passed from the SQL loader program used in data loading process 102. A plurality of stored procedures 104 are invoked by the data loading process 102, in order to run summary procedures against the transition tables 103. These summary procedures update, replace or append summary tables 105 used for administrative or reporting purposes. Summarized data tables are referred to as “fact” tables.

The fact tables 105 store information summarized from transition tables 103. Fact tables 105 store statistical information that is used for administrative and reporting purposes, and the like. A plurality of administrative procedures 106 apply business logic to client data in order to create meaningful management information. Administrative procedures 106 are used to make changes to groupings tables 107. Administrative procedures 106 comprise analytic processes that add, edit or delete records in groupings tables 107, and are executed against the fact tables 105 and administrative lookup tables 109, using categories pre-defined in the groupings tables 107. In a specific embodiment, these categories correspond to an administrative hierarchy.

Lookup tables 109 can be any of a pre-defined look up table, and a look up table defined by a client. Predefined lookup tables include, for example: a corporation, a division, a group, a product, and a source category. Client defined look up tables include: a campaign type, a campaign status, a panel type, a panel subtype, a package, a list category, a list name, a list segment, a source ID description. Lookup tables 109 are used to translate codes or values in fact tables 105, or to apply group categories in grouping tables 107.

Groupings tables 107 store information about subscription marketing campaigns and the like. Groupings tables 107 contain one or more field categories that reflect an administrative hierarchy of the client's business process. The groupings tables 107 provide a means of categorizing data related to subscription marketing campaigns and relating these groupings to statistical facts tables 105. Facts tables 105, groupings tables 107, and administrative lookup tables 109 are stored in the database server system 17. In specific embodiments, facts tables 105, groupings tables 107, and administrative lookup tables 109 comprise a single database, or may be stored separately in a plurality of databases.

Clients may use online administrative functions that interact with facts tables 105, administrative groupings tables 107, and administrative lookup tables 109 via a web browser operative on client system 14-1, which is coupled to web server system 12. Web server 12 comprises a computer and computer software that, together with the web browser operative on client 14-1, enable users to view HTML documents, and the like. Web server 12 is coupled to application server 15. Web server 12 and application server 15 can be provided by an application service provider (ASP) web-site, for example. In specific embodiments, clients may use online reporting functions that interact with facts tables 105, administrative groupings tables 107, and administrative lookup tables 109 via a web browser operative on client system 14-2, which is coupled to web server 12. Web server 12 is coupled to application server 15. Web server 12 and application server 15 can be provided by an application service provider (ASP) web-site, for example. In specific embodiments, the server systems 12, 15, and 17 that implement report functions and the server systems 12, 15, and 17 that implement administrative functions may be co-resident on a single computer system, or may be distributed among a plurality of computer systems. Web server 12 interfaces with database server system 17, that comprises a database for containing the various tables, such as facts tables 105, administrative groupings tables 107, and administrative lookup tables 109, illustrated in Fig. 3.

Fig. 4 illustrates a diagram of representative data acquisition, processing and loading in a specific embodiment according to the present invention. Fig. 4 illustrates one process for taking transaction or summary level information 101 from one or more outside sources, analyzing the data, categorizing and loading the data into database tables for further analysis and administration. The input information 101 is presented in electronic form from a variety of storage mechanisms, such as a CD, a tape, or obtained using a file transfer protocol (FTP), or the like. The information 101 relates to fulfillment and processing of subscription-based transactions. The format and level of summarization of information 101 varies among

specific embodiments, and may be customized to varying degrees for individual clients. Frequency input files are input that varies according to the client. Further, input information 101 can be obtained from a variety of sources, such as, for example a data processing vendor, or a client data source.

5           The information 101 is processed and loaded into transisiton tables 103 by a plurality of subprocesses of the data process 102 of Fig. 3. A file archival process 202, imports the information 101 and archives it into a storage repository 203. In a specific embodiment, the information 101 is organized in folders, which are organized by client and date. Other organization paradigms may be used in various specific embodiments. A data loader process 204 retrieves archived information from the repository 203. Based upon a type, location and owner of the information in the repository 203, the data loader executes processing of the information. The information is processed from repository 203 in this manner in serial fashion. The data loader 204 may be a software process written in any of a variety of languages, but in a presently preferable embodiment, data loader 204 is a java application that invokes scripts written in PERL. Once the information is processed from repository 203, a PERL DTS SQL loader transforms the information into a format compatible with storage into transition tables 103. This process also serves as a checkpoint restart mechanism. A checkpoint restart represents a rollback stage in the data loading process. If it is necessary to restore the database to an earlier version, this is accomplished by referring back to transition tables and then rerunning summary procedures. A rollback is triggered for a variety of reasons, including corrupted data feed files, corrupted summary procedures, and so forth. Then, as illustrated by Fig. 3, a plurality of stored procedures 104 are invoked to run summary procedures against the transition tables 103. These summary procedures update, replace or append summary tables 105 used for administrative or reporting purposes. Summarized data tables are referred to as “fact” tables. The facts tables 105 contain statistical information summarized from the transition tables 103 for administrative and reporting purposes.

Fig. 5A illustrates a diagram of representative online administrative system in a specific embodiment according to the present invention. Administrative functions enable a user to add, edit or change records in various database tables. In a specific embodiment illustrated by Fig. 5A, the administrative functions are provided using a browser-based interface which can be executed on a client computer, such as client system 14-1, for example. Client 14-1 interacts with web server 12 over network connection 16, which can be any of a number of network topologies, and may include the Internet, for example. Web

server system 12 connects with application server system 15, having one or more administrative processes 303 stored thereon. Administrative processes 303 can be implemented in a variety of different ways in various specific embodiments. However, in a presently preferable embodiment, administrative processes 303 are Visual Basic (VB) scripts.

Administrative processes 303 provide a variety of functionality to specific embodiments. For example, scripts written in the Visual Basic programming language provide the basis for interactive web pages. These scripts also include triggers that activate stored procedures which effect changes to database tables. Application server system 15 interfaces with database server system 17. The application server system include Visual Basic scripts, stored procedures, and security verification procedures. Application server system 15 has one or more administrative stored procedures 304 resident thereon. Administrative stored procedures 304 provide a central point for analytic processing that applies business logic to client data to create meaningful management information. Administrative procedures 304 are used to execute changes to groupings tables 107. These administrative procedures 304 comprise analytic processes that add, edit or delete records in groupings tables 107, and are executed against the facts tables 105 and administrative lookup tables 109, using categories pre-defined in the groupings tables 107. These categories correspond to an administrative hierarchy. Clients interact with administrative procedures 304 via the application server 15 which can be connected to an application service provider's (ASP) web-site, for example. The application server 15 activates various unique and proprietary stored procedures. The administrative procedures 304 provide the capability to update grouping categories corresponding to an administrative hierarchy in the groupings table 107 and user defined categories in the administrative tables 109.

Fig. 5B illustrates a detail diagram of a representative online administrative system of Fig. 5A in a specific embodiment according to the present invention. The administrative online system comprising administrative processes 303, application server system 15, administrative procedures 304, and database server system 17. Information in the groupings table 107 comprises a grouping 402 entitled "Groupings Tables:" that comprises a plurality of field categories, including a source, a campaign, a panel and a key. It is noteworthy that some specific embodiments may not have all of the field categories shown, and/or may comprise other field categories not illustrated by Fig. 5B. Administration table 109 comprises a grouping 404 entitled "User-Administered Lookup Tables:" which comprises a plurality of field categories, as illustrated by Fig. 5B. It is noteworthy that some specific embodiments may not have all of the field categories shown, and/or may comprise

other field categories not illustrated by Fig. 5B. In a specific embodiment, users can create custom grouping categories in order to analyze the data and customize reporting.

Administrative procedures 304 comprises a plurality of user triggered stored procedures 406 that enable users to add or edit items into grouping tables 107. Further, administrative  
5 procedures 304 comprises a plurality of user triggered stored procedures 408 that enable users to customize administration tables 109.

Fig. 6 illustrates an administrative framework that provides a grouping hierarchy in a specific embodiment according to the present invention. A grouping is a unique, proprietary categorization of information. Fig. 6 illustrates the administrative  
10 framework underlying Figs. 5A-5B, which provides a basis for administrative groupings, as well as enabling the user to create custom grouping categories in order to gain further insight and customize reports. Administrative frameworks support insights into a client's business process, which in turn provide a consistent means for reporting subscription marketing information, as well. Fig. 6 illustrates a representative groupings hierarchy 410 having a  
15 plurality of information categories. A Corporation 412 is a pre-defined grouping that denotes a company. Corporate values are stored in a corp\_info database table in the facts tables database 105. A Division 414 is a pre-defined grouping that denotes a division within a company. Division values are stored in the corp\_info database table. A Group 416 is a pre-defined grouping that denotes a sub-division of a division. Group values are stored in the  
20 corp\_info database table. A Product 418 is a pre-defined grouping that denotes a specific product sold. Product values are stored in the corp\_info database table. A Source Category 420 is a pre-defined grouping that denotes a category of business within a product (i.e. Agent, Billing, DTP, Gifts, Renewals). Source Category values are stored in a src\_category database table in the facts tables database 105. A Source 422 is a user-defined grouping that denotes a  
25 channel of business within a source category grouping 420. A business channel is a group of campaigns. Source values are stored in a source database table in the groupings tables database 107, and propagated to all lower-level database tables. A Campaign 424 is a user-defined grouping that denotes a particular promotion effort or event. Campaign values are stored in a campaigns database table in groupings tables database 107, and propagated to all  
30 lower-level database tables. A Campaign Type 442 is a user-defined grouping that denotes a category of Campaigns. Campaign Type values are stored in the campaigns database table, and propagated to all lower-level database tables. A Campaign Status 444 is a user-defined grouping that denotes a category of Campaigns. Campaign Type values are stored in the campaigns database table, and propagated to all lower-level database tables. A Panel

(campaign subgroup) 426 is a user-defined grouping that denotes a campaign subgroup (a grouping of keys). Panel values are stored in a panels database table in groupings tables database 107, and propagated to all lower-level database tables. A Panel Type 432 is a user-defined grouping that denotes a category of panels. Panel Type values are stored in the panels database table, and propagated to all lower-level database tables. A Panel Subtype 434 is a user-defined grouping that denotes a category of panels. Panel Subtype values are stored in the panels database table, and propagated to all lower-level database tables. A Package 436 is a user-defined grouping that denotes a category of panels or keys. Package values are stored in the panels database table, and propagated to all lower-level database tables. A Key 428 is an identifier that denotes a grouping of individuals who are sent a common promotion effort. Key values are stored in a keys database table in groupings tables database 107. A List Category 452 is a user-defined grouping that denotes a broad category or promotional lists. List Category values are stored in the keys database table. A promotional list is a specific source for names used to promote products and services. A List Name 454 is a user-defined grouping that denotes a specific source for names used to promote products and services. List Name values are stored in the keys database table. A List Segment 456 is a user-defined grouping that denotes a specific selection from a list. Selections are often based on demographic or behavioral characteristics (i.e., age, sex, region). List Segment values are stored in the keys database table. Fig. 6 is merely one example of an administrative framework. Many other administrative frameworks that support insights into a client's business process can be readily generated by those of ordinary skill in the art based upon the teachings of the present invention. Such administrative frameworks provide a consistent means for reporting subscription marketing information, as will be discussed in further detail below.

Fig. 7 illustrates a diagram of representative reporting process in a specific embodiment. The reporting functions enable users to select reports and query database tables in order produce subscription marketing reports. Reporting functions enable users to selectively choose reports and query database tables in order produce useful subscription marketing reports. In a specific embodiment, the reporting functions are provided through a browser-based interface. Reports provide structured, consistent information that describes the business performance of marketing campaigns. Fig. 7 illustrates a report generating framework, comprising a browser-based interface which can be executed on a client computer, such as client system 14-2, for example. Client 14-2 interacts with web server 12 over network connection 16, which can be any of a number of network topologies, and may



include the Internet, for example. Web server system 12 connects with application server system 15, having one or more report processes 501 stored thereon. Report processes 501 can be implemented in a variety of different ways in various specific embodiments.

However, in a presently preferable embodiment, report processes 501 are script programs written in the Visual Basic programming language, which provide the basis for interactive web pages displaying the reports.

Application server system 15 connects to a database server system 17 that comprises database management functions, such as locating requested data records, and the like. Database server system 17 manages a report catalog table 503, comprising report file settings such as; report name, report alias, location, and related application settings. Database server system 17 also manages a saved report table 505 that stores information about saved reports. A saved report record includes report selection and location, selection query string, and parameters including user-defined report name, report group and comments. After a specific report file is selected by a user at client 14-2, the selection is communicated through network 16 to web server 12, which passes the selection to application server system 15. The application server system 15 queries the database server 17 to obtain the selected report from report catalog table 503, or from stored report table 505. Then, a valid query string is passed to the report in a report execution process 502. Then, the report is rendered by a report rendering process 504. Reports may be rendered in a variety of ways in various specific embodiments, however, in a presently preferred embodiment, the report is rendered through the browser via an ActiveX control viewer. The viewer presents a picture of grouping and statistical information according to a pre-defined format. Users control the content of the report by limiting the selection of data.

Fig. 8 illustrates a diagram showing a representative hyperlink chain linking reports in a specific embodiment according to the present invention. Reports are hyperlinked to other reports, web-pages, or files. The hyperlink chain enables users to launch reports from within a report, or view related files or web-pages, thus reducing the amount of time required to access additional views of information. Fig. 8 illustrates a report 10 comprising a file, for example, that displays grouping and statistical information in a specific format. Users control the content of report 10 from a browser at client 14-2 by limiting the selection of data that is incorporated into the report. The selection of data is accomplished using a reporting online functional application. Report 10 comprises a first plurality of hypertext links 600, including a link to supplemental report 602 that provides a hyperlink to a process that renders a report file that displays grouping and statistical information in a specific

format. Data selection criteria is automatically passed to supplemental reports based on the origin of the hyperlink. Hypertext links 600 also includes a link to web page 604 that provides a hyperlink to a process that renders a web page, a link to file 606 that provides a hyperlink to a process that renders or executes a file, a link to an export to file 608 that provides a function that exports the grouping and statistical values presented in the ActiveX report viewer to a file (txt, xls, html). A second plurality of hypertext links 610 is also associated with report 10. Hypertext links 610 comprises links 612-618, which are analogous to links 602-608 of hypertext links 600.

Fig. 9 illustrates a diagram of a representative access security for a database and reporting system in a specific embodiment according to the present invention. Security techniques enable specific embodiments to prevent unauthorized access to client data. In a specific embodiment, secure users are provided with usernames and passwords. Access rights are set by restricting access to specific database tables depending on access security level settings, which are also associated with unique combinations of usernames and passwords. The security functions are provided to users of a browser-based interface which can be executed on a client computer, such as client system 14-1, for example. Client 14-1 interacts with web server 12 over network connection 16, which can be any of a number of network topologies, and may include the Internet, for example. Web server system 12 connects with application server system 15, having one or more administrative processes 303 and reporting processes 501 stored thereon. Application server system 15 has access to a plurality of user security access tables 711, which store user access information, including username, password, database access rights settings and administrative functions rights settings, for example. Application server system 15 comprises stored procedures and security verification procedures, which can be implemented in a variety of different ways, but in a specific embodiment are Visual Basic scripts. A security verification procedure executing in application server system 15 performs a verification of a user's access authority 704. If the user has access rights to the administrative and/or reporting functions, which can be determined by matching a username and a password against user security access tables 711, the user is permitted to access the administrative and/or reporting functions.

However, if a mismatch in username and/or password occurs, then a verification failure process 702 returns the user to a web server logon page in order to provide another opportunity to submit a valid username and password. In a specific embodiment, the above described access control to the administrative and/or reporting functions of the application server 15 may be used in conjunction with access control provided by the web

server 12 to provide two layer security functionality. Further, in a specific embodiment, the above described access control to the administrative and/or reporting functions of the application server 15 may be implemented using a plurality of access levels. In a representative specific embodiment, a first security level provides access to basic functions, such as username and password selection and database query functions. A second security level provides access to administrative functions, such as setting permissions for database tables and the like.

Fig. 10 illustrates a diagram of a representative process for denormalizing database tables in a specific embodiment according to the present invention. In accordance with the invention, database tables are denormalized by propagating information from one table to another, but there is no relationship between the tables by way of a database join operation. In a specific embodiment, this function is provided by specific procedures of the stored procedures 106. The specific stored procedures are activated by users using a browser program at client 14-1, for example, in cooperation with an administrative application. The denormalized database table enables report generation to be performed more quickly by reducing overhead associated with joining a plurality of database tables.

Denormalization takes as input a plurality of tables, from the facts tables 105 and lookup tables 109 and produces a key table 1009, which contains a complete set of grouping information, in groupings tables 107. The groupings (keys) table 107 is the basis for many of the reporting features available in specific embodiments. The groupings table 107 is populated, updated and appended in a variety of ways. For example, a user may apply updates to groupings table 107 using the online administrative interface from a browser program on client 14-1. Another technique updates grouping table 107 automatically by referencing a set of user administered lookup tables 109. A yet further technique updates grouping table 107 automatically by referencing a set of lookup tables 109 maintained by a database administrator. The groupings table 107 stores the information used to create a variety of marketing reports corresponding to the administrative framework (groupings hierarchy), without having dependencies on other database tables. Specifically, the key table 1009 in Fig. 10 is used for report generation in a process which matches groupings in the key table 1009.

Fig. 10 illustrates a representative denormalization process, in which a source category table 1001, which is resident in administrative tables 109, provides information to a source table 1003, which is also resident in administrative tables 109, by an update source table process 1002. An update campaign table process 1004 reflects information from source

table 1003 to a campaign table 1005. An update panels table process 1006 propagates information from campaign table 1005 to a panels table 1007. An update keys table process 1008 updates keys table 1009, in groupings tables 107 based upon the contents of panels table 1007. In a specific embodiment, update source table process 1002, update campaign table process 1004, update panels table process 1006, and update keys table process 1008 are implemented as stored procedures. By carrying information from one database table to another, these procedures effectively link tables in different databases without requiring an database table join operation.

A representative example of the information exchanged from one database table to another during denormalization processing follows:

---

Source Category Table to Source Table

product\_id  
src\_cat\_name (source category name)

Source Table to Campaign Table

product\_id  
src\_cat\_name (source category name)  
source (source name)

Campaign Table to Panel Table

product\_id  
src\_cat\_name (source category name)  
source (source name)  
camp\_name (campaign name)  
mail\_date\_fr (mail date)  
camp\_status (campaign status)  
e\_set\_name (expense set name)  
r\_set\_name (revenue set name)  
budgeted\_vol (budgeted volume)  
budgeted\_gross\_pct (budgeted gross response percent)

Panel Table to Keys Table

product\_id  
src\_cat\_name (source category name)  
source (source name)  
camp\_name (campaign name)  
mail\_date\_fr (mail date)  
camp\_status (campaign status)  
e\_set\_name (expense set name)  
r\_set\_name (revenue set name)  
budgeted\_vol (budgeted volume)  
budgeted\_gross\_pct (budgeted gross response percent)  
panel\_name (panel name)  
panel\_start\_date (panel mail date)  
panel\_type (panel type)  
panel\_subtype (panel subtype)  
panel\_pkg (package description)

Keys Table Inputs and Overrides

All the field values listed above from the Panels Table plus:  
 key\_desc (source key description)  
 mail\_qty\_override (mail qty override qty)  
 newsstand\_sale (newsstand sale qty)  
 newstand\_sale\_marker (subtotal marker – newstand only)  
 list\_cat\_name (list category name)  
 list\_name (list name)  
 list\_segement\_name (list segment name)  
 list\_cost\_rollout\_cpm (list rollout cost per thousand)  
 list\_cost\_basis (basis for calculation list costs)  
 list\_cost\_factor (factor applied against list costs)  
 merge\_purge\_qty (merge purge qty)  
 subtotal\_flag (subtotal marker)  
 optional\_effort\_id (effort sequence marker)

Fig. 11 illustrates a diagram of a representative process for optimizing report

generation in a specific embodiment according to the present invention. According to the invention, reports may be optimized for performance or other reasons by applying

denormalizing processes, such as described above with reference to Fig. 10, to one or more of facts tables 105 and lookup tables 109, for example. Fig. 11 illustrates a process for optimizing report 10 in accordance with the invention. Fig. 11 illustrates facts tables 105 and lookup tables 109 which provide tables that serve as input to the denormalization processing described with reference to Fig. 10. Facts table 105 comprises one or more report specific fact tables 902. Lookup table 109 comprises one or more report specific lookup tables 904. Denormalization is applied to report specific fact tables 902 and report specific lookup tables 904 to produce one or more denormalized report specific groupings tables 906 in groupings tables 107. Report 10 is created from the denormalized report specific groupings tables 906 in groupings tables 107. In specific embodiments according to the present invention, this denormalization and report optimizing processing provides improved reporting capabilities by reducing at least one of a number of joins of database tables that must be made and a number of records that must be reported. These reductions enable specific embodiments to provide reports more quickly than conventional techniques. Specific embodiments deliver reports over computer networks, such as the Internet. Further, specific embodiments can produce a report from a relatively large number of transactions within a reasonable delay time.

Although specific embodiments of the invention have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the invention. The described invention is not restricted to operation within certain specific data processing environments, but is free to operate within a plurality

of data processing environments. For example, the present invention may be used to extract and store information for any domain or industry which benefits from the information extraction and storage. Additionally, although the present invention has been described using a particular series of transactions and steps, it should be apparent to those skilled in the art  
5 that the scope of the present invention is not limited to the described series of transactions and steps.

Further, while the present invention has been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present invention. The present  
10 invention may be implemented only in hardware or only in software or using combinations thereof.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made thereunto without  
15 departing from the broader spirit and scope of the invention as set forth in the claims.

WHAT IS CLAIMED IS:

1                    1.        A method for analyzing subscription information, comprising:  
2                    receiving a plurality of subscription information;  
3                    archiving said plurality of subscription information into a repository;  
4                    retrieving from said repository a plurality of selected subscription information;  
5                    transforming said plurality of selected subscription information into transition  
6 table format information;  
7                    populating a facts database with said transition format information to produce  
8 at least one of a plurality of facts tables with a summarization of said transition format  
9 information;  
10                    denormalizing said summarization of said transition format information to  
11 produce a plurality of denormalized information; and  
12                    providing a report based upon said denormalized information.

1                    2        The method of claim 1, wherein denormalization comprises:  
2                    populating selected information from at least one of a plurality of facts tables  
3 to a key table.

1                    3.        The method of claim 2, wherein said facts tables are stored in a facts  
2 table database.

1                    4.        The method of claim 3, wherein said key table is stored in a grouping  
2 table database.

1                    5.        The method of claim 2, said at least one of a plurality of facts tables  
2 comprises at least one of  
3 source category table, a source table, a campaign table, a panels table.

1                    6.        The method of claim 1, further comprising:  
2 receiving input of at least one criteria for at least one of a plurality of reports;  
3 and  
4 providing said report based upon said denormalized information according to  
5 said at least one criteria.

1                    7.        The method of claim 1, further comprising:

2 receiving input of at least one criteria for at least one of a plurality of tables;  
3 and  
4 configuring said at least one of said plurality of tables based upon said at least  
5 one criteria.

1 8. The method of claim 1, further comprising:  
2 receiving input of at least one of a username and a password;  
3 verifying whether said input of said at least one of a username and a password  
4 corresponds to a user authorized to access reporting procedures; and  
5 providing access to reporting procedures to said user if said user is determined  
6 to be authorized.

1 9. The method of claim 8, further comprising:  
2 verifying whether said input of said at least one of a username and a password  
3 corresponds to a user authorized to access administrative procedures; and  
4 providing access to administrative procedures to said user if said user is  
5 determined to be authorized.

1 10. A method for analyzing subscription information, comprising:  
2 receiving a plurality of subscription information;  
3 archiving said plurality of subscription information into a repository;  
4 retrieving from said repository a plurality of selected subscription information;  
5 transforming said plurality of selected subscription information into transition  
6 table format information;  
7 populating a facts database with said transition format information to produce  
8 at least one of a plurality of facts tables with a summarization of said transition format  
9 information;  
10 denormalizing said summarization of said transition format information to  
11 produce a plurality of denormalized information, said denormalization comprising populating  
12 selected information from at least one of a plurality of facts tables, stored in a facts table  
13 database, to a key table, stored in a grouping tables database; and  
14 providing a report based upon said denormalized information;  
15 wherein said at least one of a plurality of facts tables comprises at least one of  
16 source category table, a source table, a campaign table, and a panels table.



1            11.    The method of claim 10, further comprising:  
2            receiving input of at least one criteria for at least one of a plurality of reports;  
3    and  
4            providing said report based upon said denormalized information according to  
5    said at least one criteria.

1            12.    The method of claim 10, further comprising:  
2            receiving input of at least one criteria for at least one of a plurality of tables;  
3    and  
4            configuring said at least one of said plurality of tables based upon said at least  
5    one criteria.

1            13.    The method of claim 10, further comprising:  
2            receiving input of at least one of a username and a password;  
3            verifying whether said input of said at least one of a username and a password  
4    corresponds to a user authorized to access reporting procedures; and  
5            providing access to reporting procedures to said user if said user is determined  
6    to be authorized.

1            14.    The method of claim 13, further comprising:  
2            verifying whether said input of said at least one of a username and a password  
3    corresponds to a user authorized to access administrative procedures; and  
4            providing access to administrative procedures to said user if said user is  
5    determined to be authorized.

1            15.    A computer program product for analyzing subscription information,  
2    said computer program product comprising:  
3            code that receives a plurality of subscription information;  
4            code that archives said plurality of subscription information into a repository;  
5            code that retrieves from said repository a plurality of selected subscription  
6    information;  
7            code that transforms said plurality of selected subscription information into  
8    transition table format information;

code that populates a facts database with said transition format information to produce at least one of a plurality of facts tables with a summarization of said transition format information;

code that denormalizes said summarization of said transition format information to produce a plurality of denormalized information, said denormalization comprising populating selected information from at least one of a plurality of facts tables, stored in a facts table database, to a key table, stored in a grouping tables database;

code that provides a report based upon said denormalized information; and  
a computer readable storage medium for holding the codes.

16. The computer program product of claim 15, further comprising:  
code that receives input of at least one criteria for at least one of a plurality of reports; and

code that provides said report based upon said denormalized information according to said at least one criteria.

17. The computer program product of claim 15, further comprising:  
code that receives input of at least one criteria for at least one of a plurality of tables; and

code that configures said at least one of said plurality of tables based upon said at least one criteria.

18. The computer program product of claim 15, further comprising:  
code that receives input of at least one of a username and a password;  
code that verifies whether said input of said at least one of a username and a password corresponds to a user authorized to access reporting procedures; and

code that provides access to reporting procedures to said user if said user is determined to be authorized.

19. The computer program product of claim 18, further comprising:  
code that verifies whether said input of said at least one of a username and a password corresponds to a user authorized to access administrative procedures; and

code that provides access to administrative procedures to said user if said user is determined to be authorized.

20. An apparatus for analyzing subscription information, comprising:

means for receiving a plurality of subscription information;  
 means for archiving said plurality of subscription information into a  
 repository;  
 means for retrieving from said repository a plurality of selected subscription  
 information;  
 means for transforming said plurality of selected subscription information into  
 transition table format information;  
 means for populating a facts database with said transition format information  
 to produce at least one of a plurality of facts tables with a summarization of said transition  
 format information;  
 means for denormalizing said summarization of said transition format  
 information to produce a plurality of denormalized information, said denormalization  
 comprising populating selected information from at least one of a plurality of facts tables,  
 stored in a facts table database, to a key table, stored in a grouping tables database; and  
 means for providing a report based upon said denormalized information.

21. A system for analyzing subscription information, comprising:  
 at least one of a plurality of clients;  
 a web server; connected to said at least one of a plurality of clients by a  
 computer network;  
 an application server;  
 a database server; connected to said application server and said web server by  
 a communication medium, wherein said at least one client receives a plurality of subscription  
 information and forwards said plurality of subscription information via said web server and  
 said application server to said database server, which archives said plurality of subscription  
 information into a repository; and wherein  
 said application server retrieves from said repository a plurality of selected  
 subscription information and transforms said plurality of selected subscription information  
 into transition table format information; and wherein  
 said database server populates a facts database with said transition format  
 information to produce at least one of a plurality of facts tables with a summarization of said  
 transition format information; and wherein  
 said application server denormalizes said summarization of said transition  
 format information to produce a plurality of denormalized information, said denormalization

19 comprising populating selected information from at least one of a plurality of facts tables,  
20 stored in a facts table database, to a key table, stored in a grouping tables database; and  
21 wherein  
22 said application server provides a report based upon said denormalized  
23 information.

1 22. The system of claim 21, wherein said at least one of a plurality of  
2 clients receives input of at least one criteria for at least one of a plurality of reports; and  
3 forwards said input via said web server to said application server, which provides said report  
4 based upon said denormalized information according to said at least one criteria.

1 23. The system of claim 21, wherein said at least one of a plurality of  
2 clients receives input of at least one criteria for at least one of a plurality of tables; and  
3 forwards said input via said web server to said application server, which configures said at  
4 least one of said plurality of tables based upon said at least one criteria.

1 24. The system of claim 21, wherein said at least one of a plurality of  
2 clients receives input of at least one of a username and a password; and forwards said input  
3 via said web server to said application server, which verifies whether said input of said at  
4 least one of a username and a password corresponds to a user authorized to access reporting  
5 procedures; and provides access to reporting procedures to said user if said user is determined  
6 to be authorized.

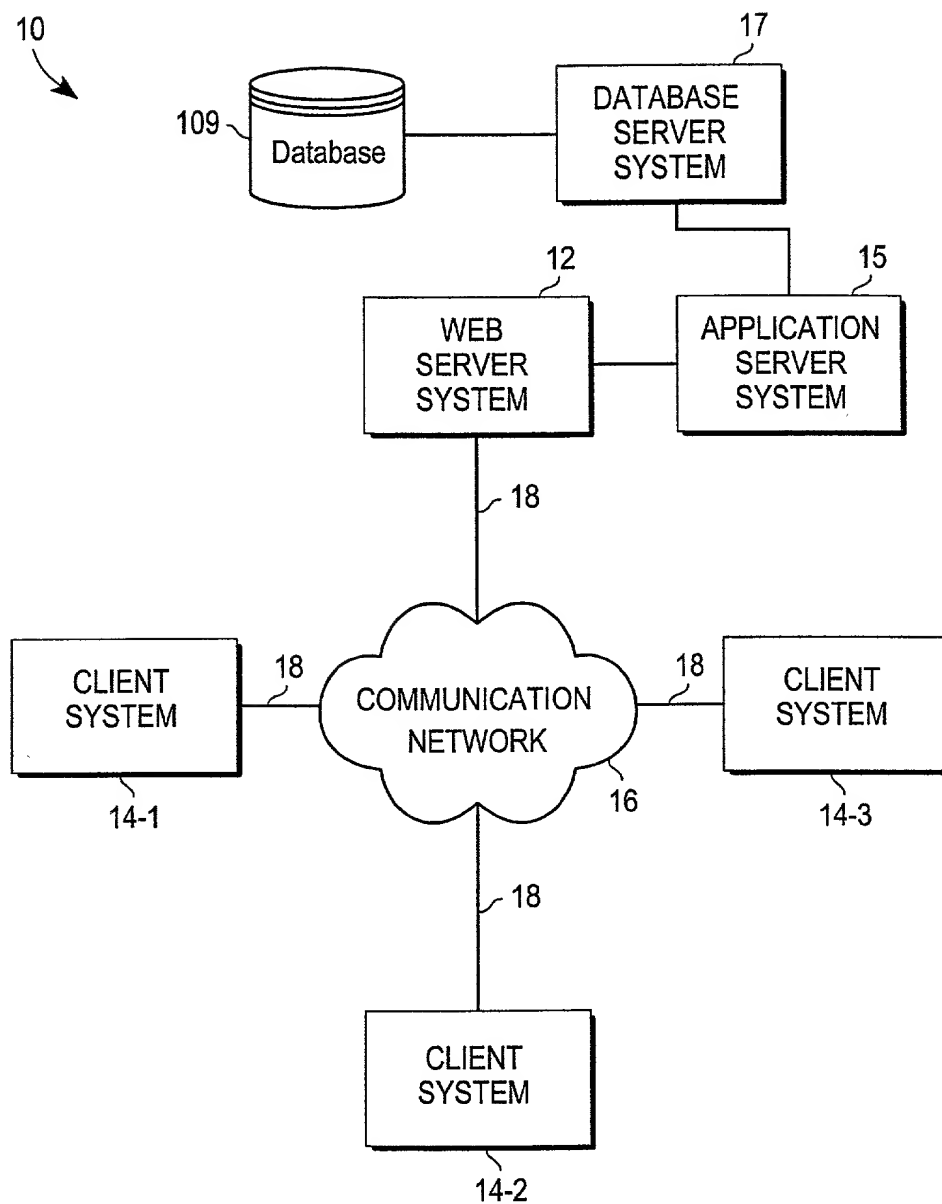
1 25. The system of claim 24, wherein said application server verifies  
2 whether said input of said at least one of a username and a password corresponds to a user  
3 authorized to access administrative procedures; and provides access to administrative  
4 procedures to said user if said user is determined to be authorized.

## **SUBSCRIPTION MEMBERSHIP MARKETING APPLICATION FOR THE INTERNET**

### ABSTRACT OF THE DISCLOSURE

The present invention provides network based reporting techniques designed to meet the needs of subscription based marketers. Clients access, edit report on, and analyze their subscriber database using the Internet, for example. In a representative embodiment, a plurality of client computers connect to the Internet using a local area network (LAN), for example. The client computers pass security protocols before gaining access to web, application and database servers. Once connected, the client computers are able to retrieve and display information using a web browser. Database contents are denormalized to provide the capability to generate reports based upon relatively large amounts of information. In addition, the data is transformed into a structure specifically to facilitate and enhance multi-channel subscription marketing activities.

PA 3102715 v3

*Fig. 1*

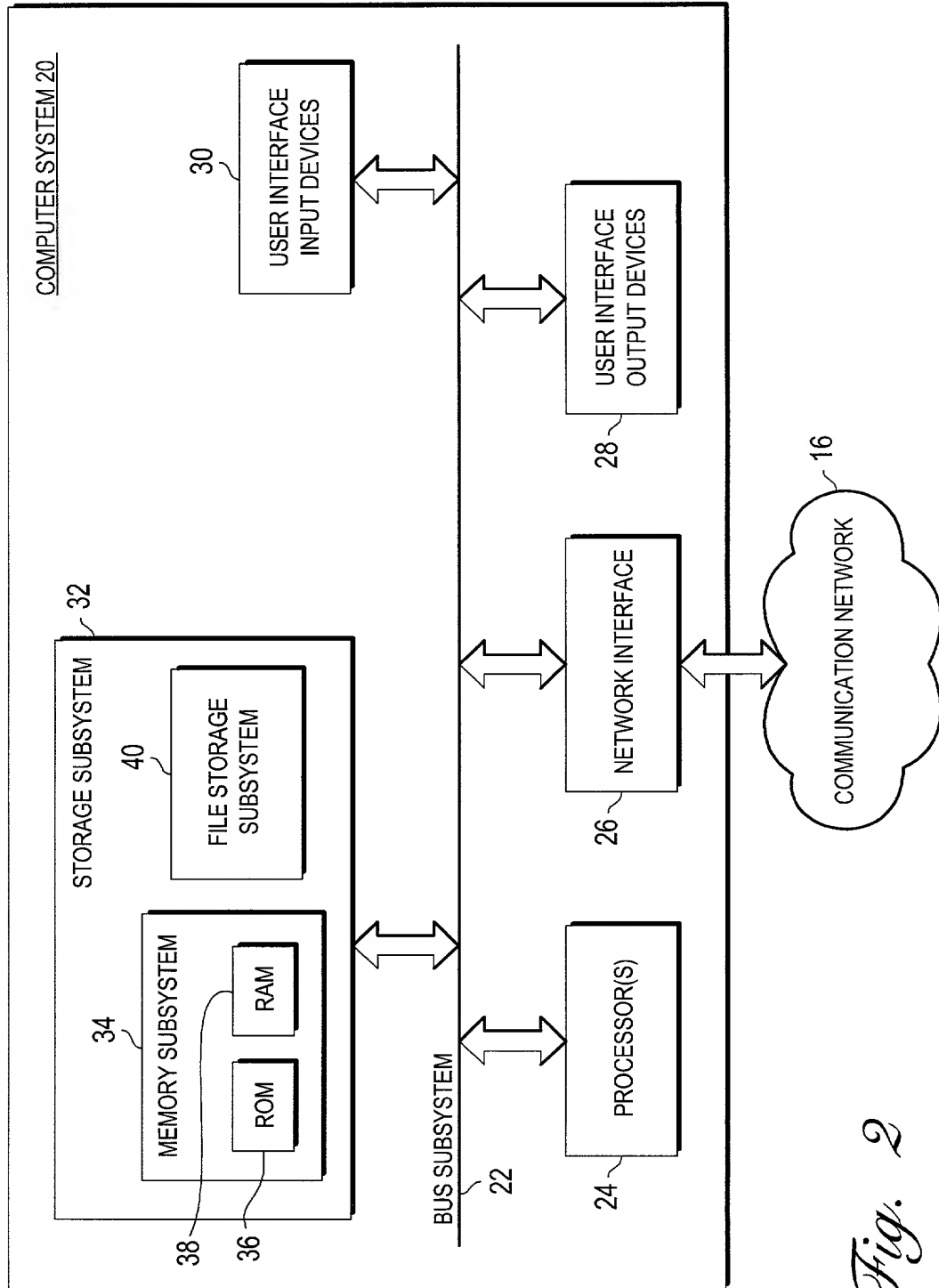
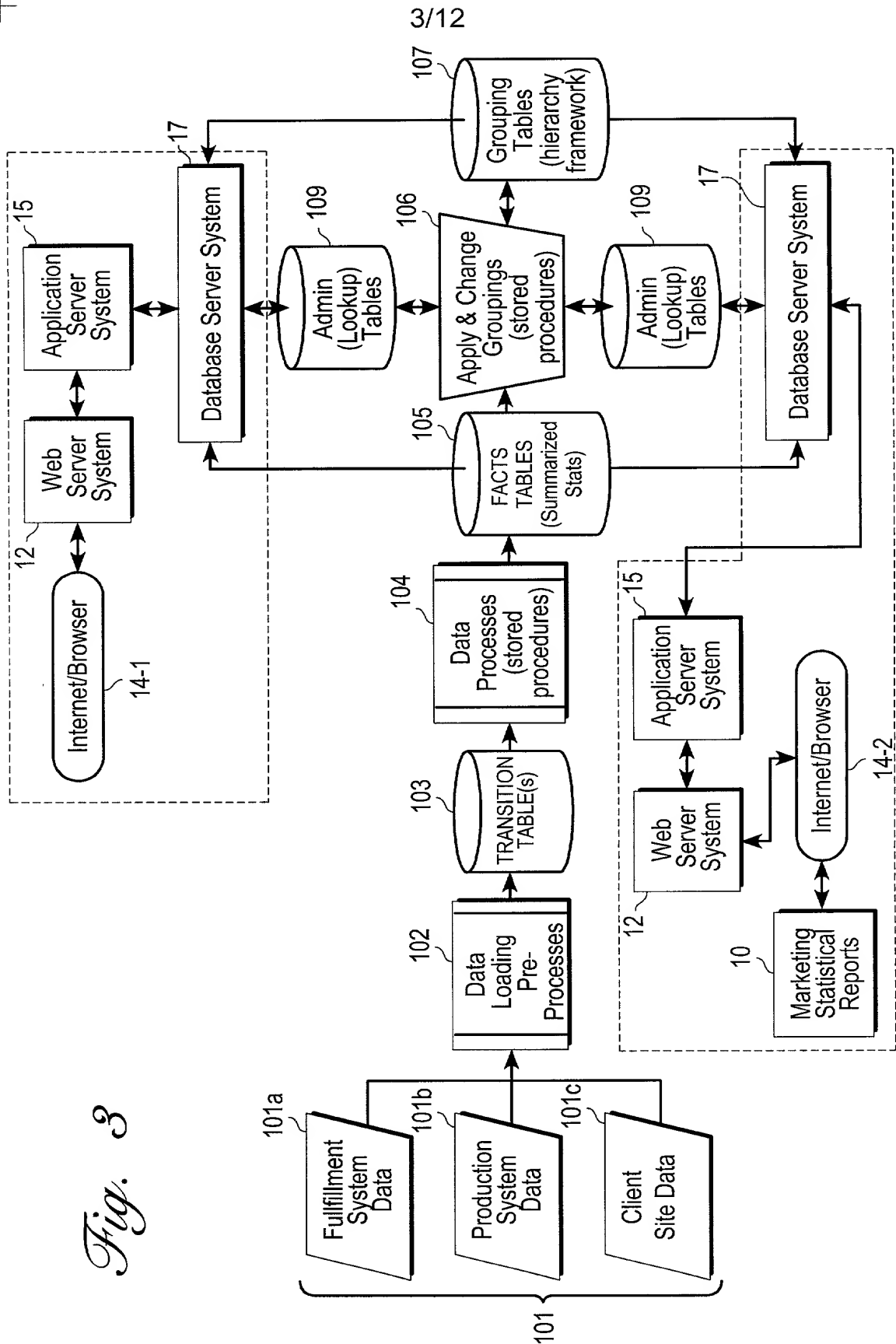


Fig. 3





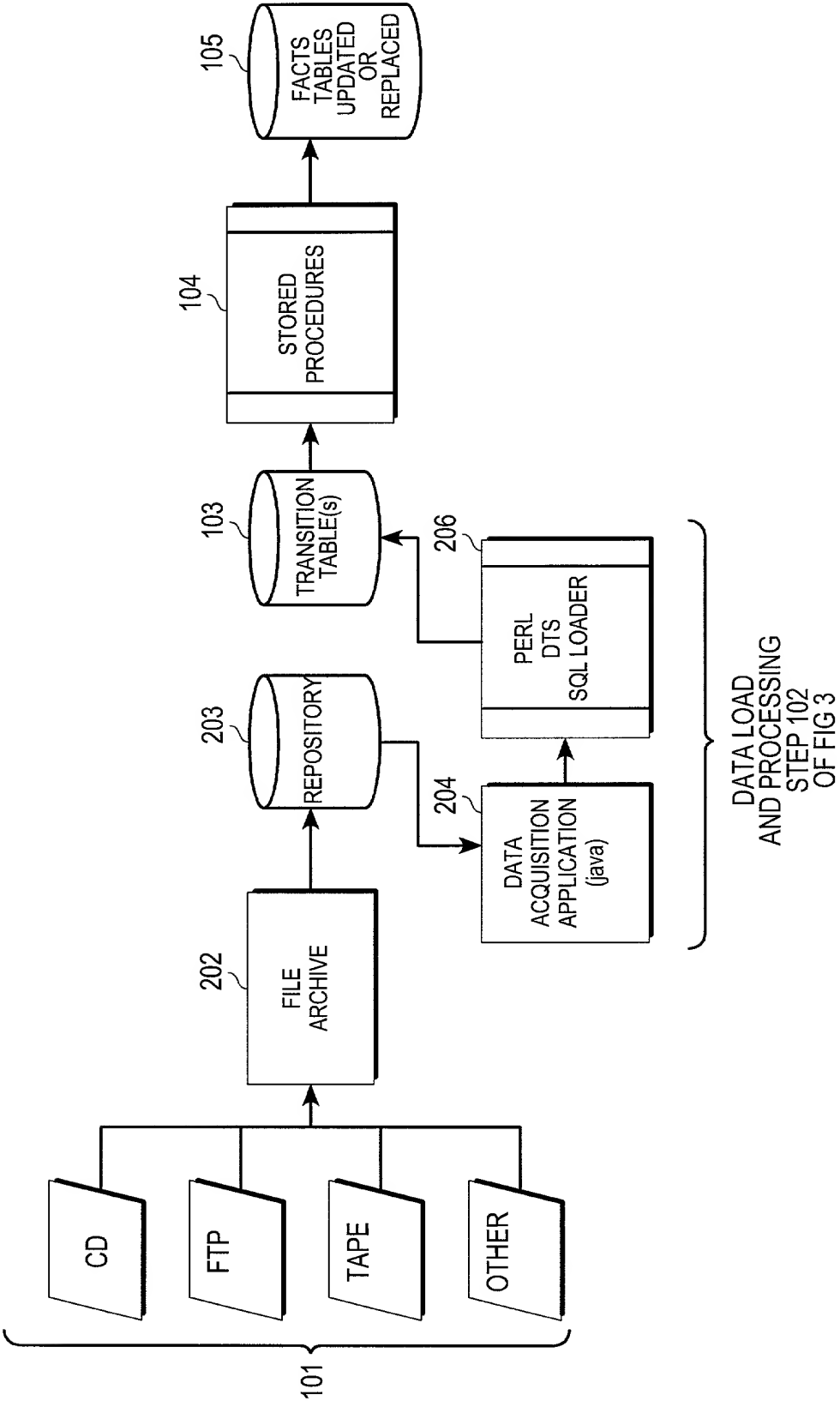
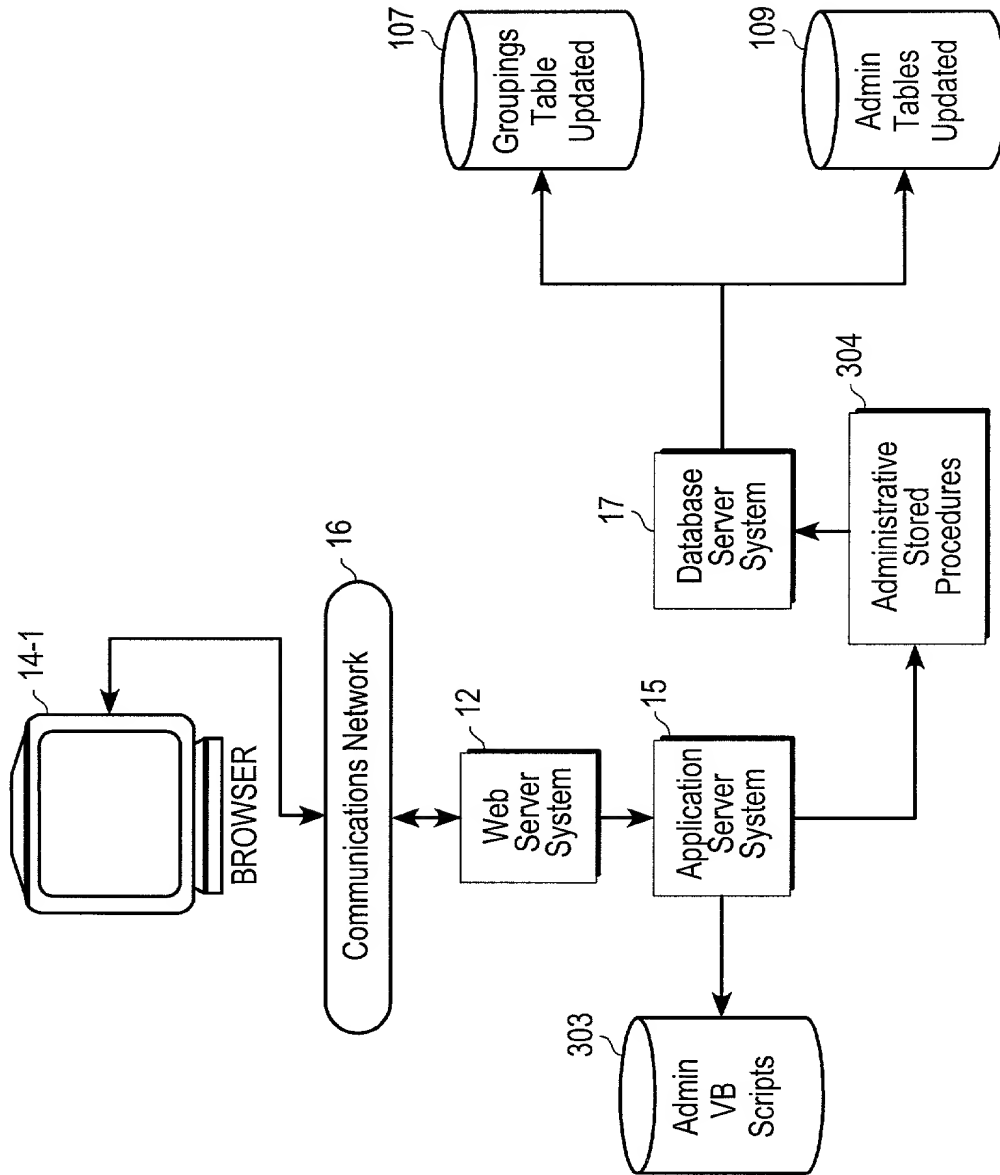


Fig. 4

*Fig. 5A*

+

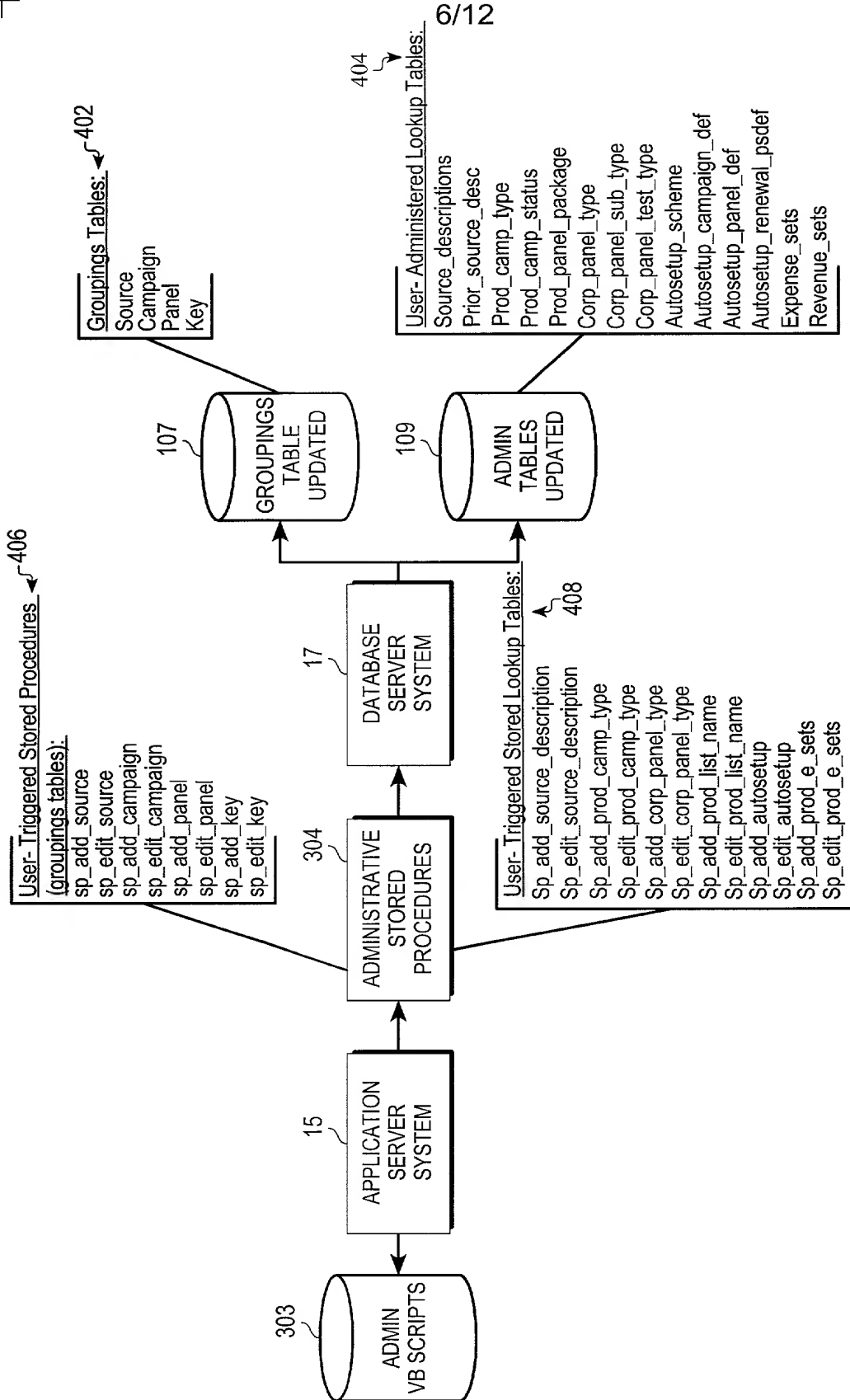
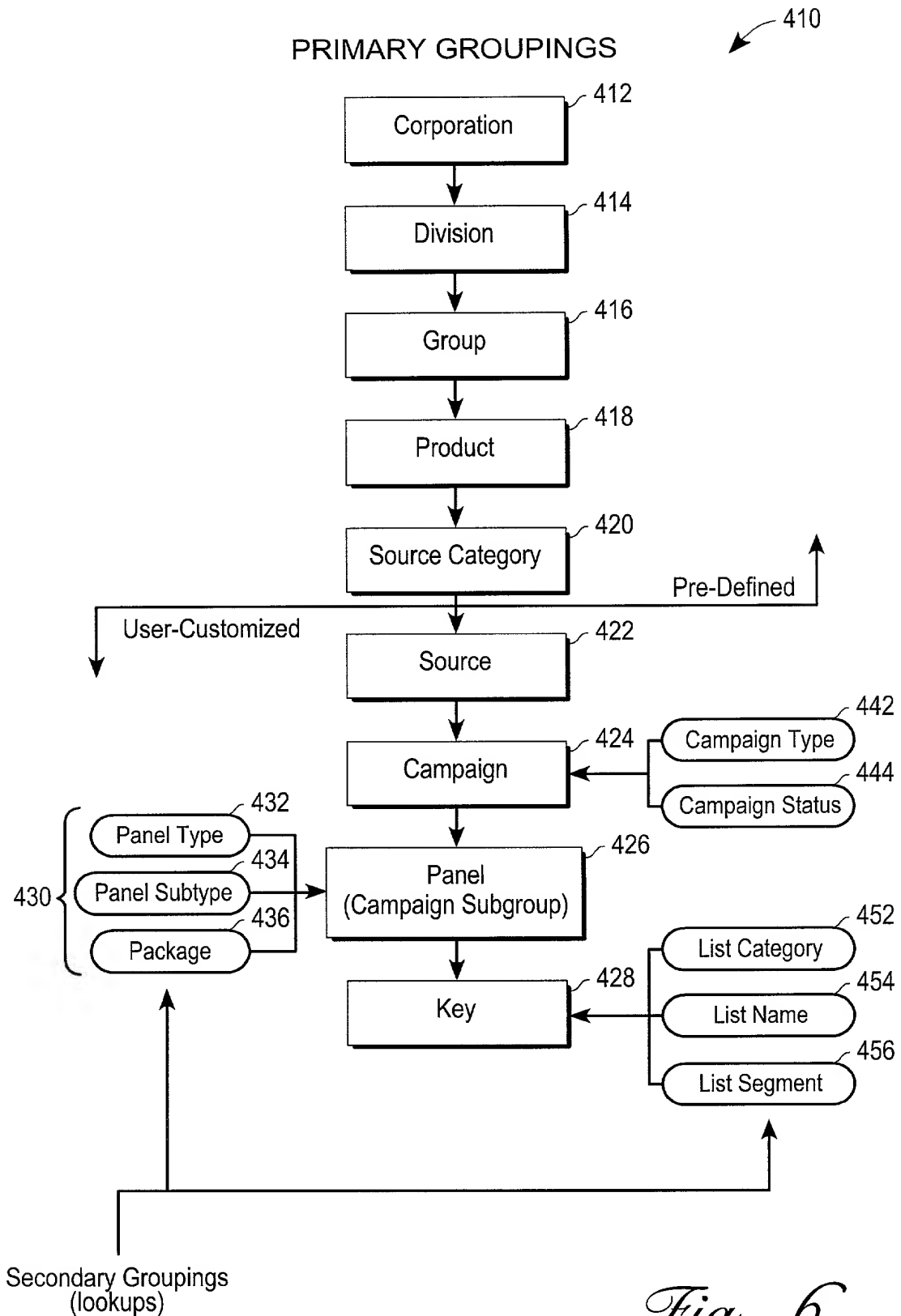
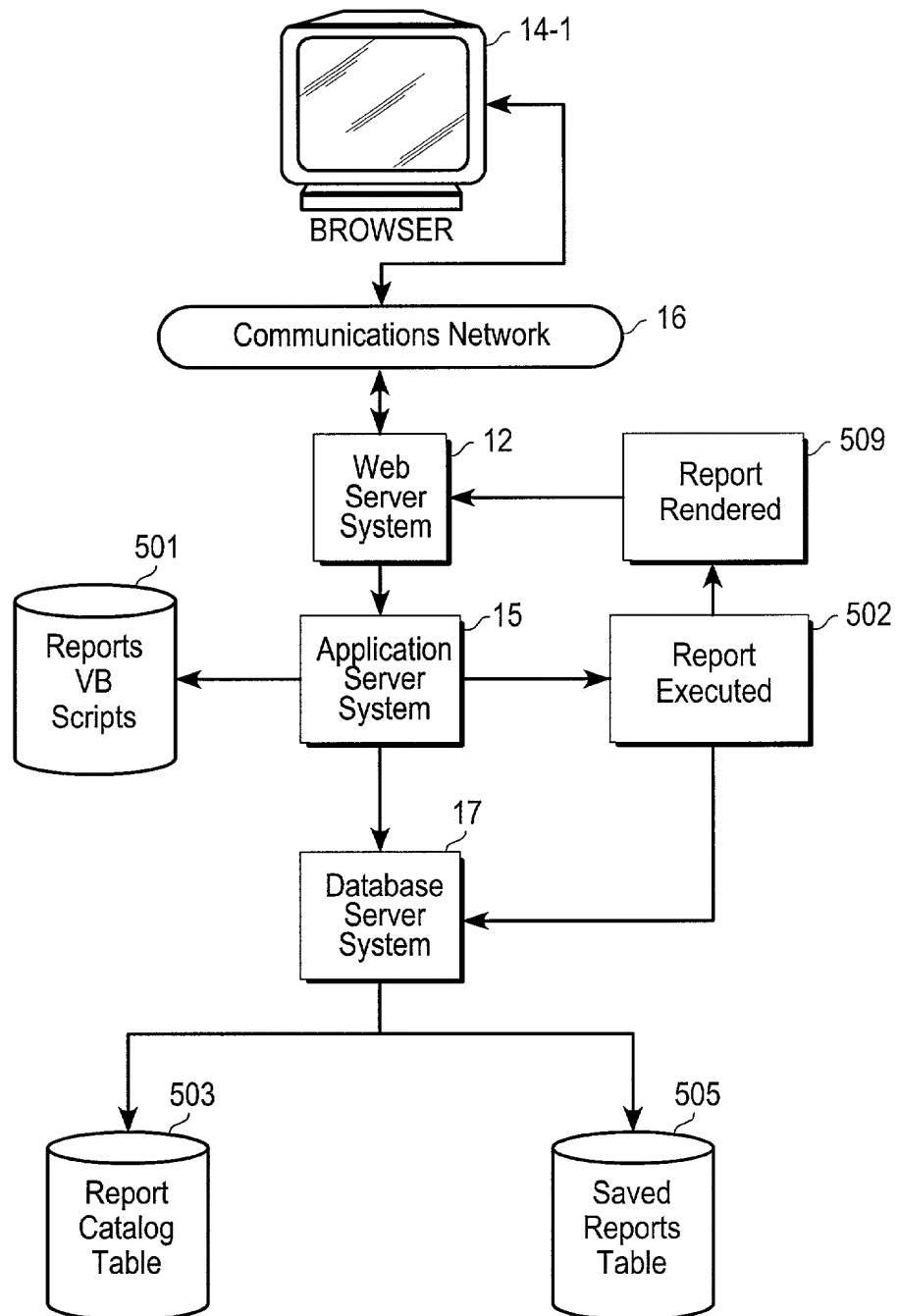


Fig. 5B

+

7/12



*Fig. 7*

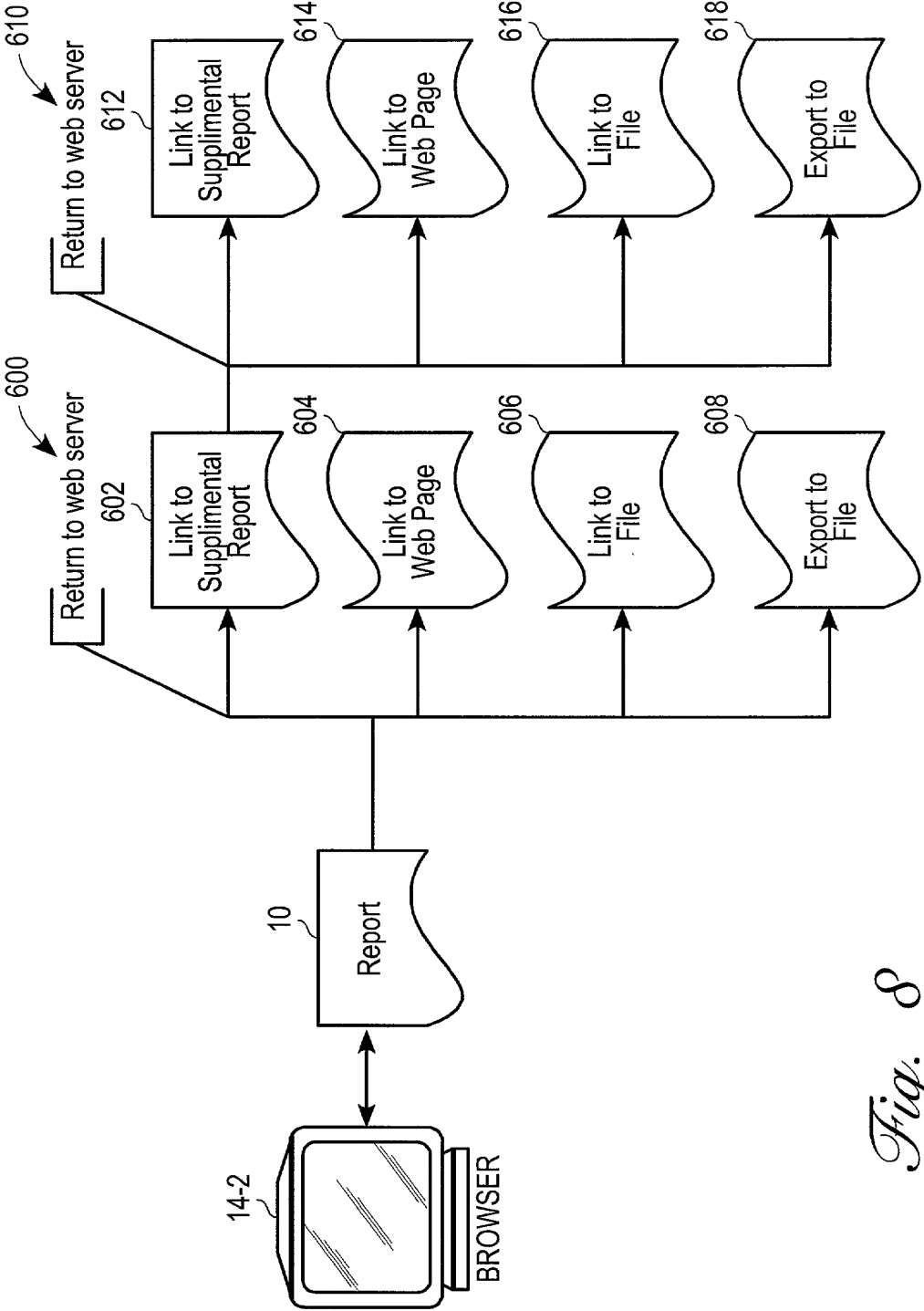
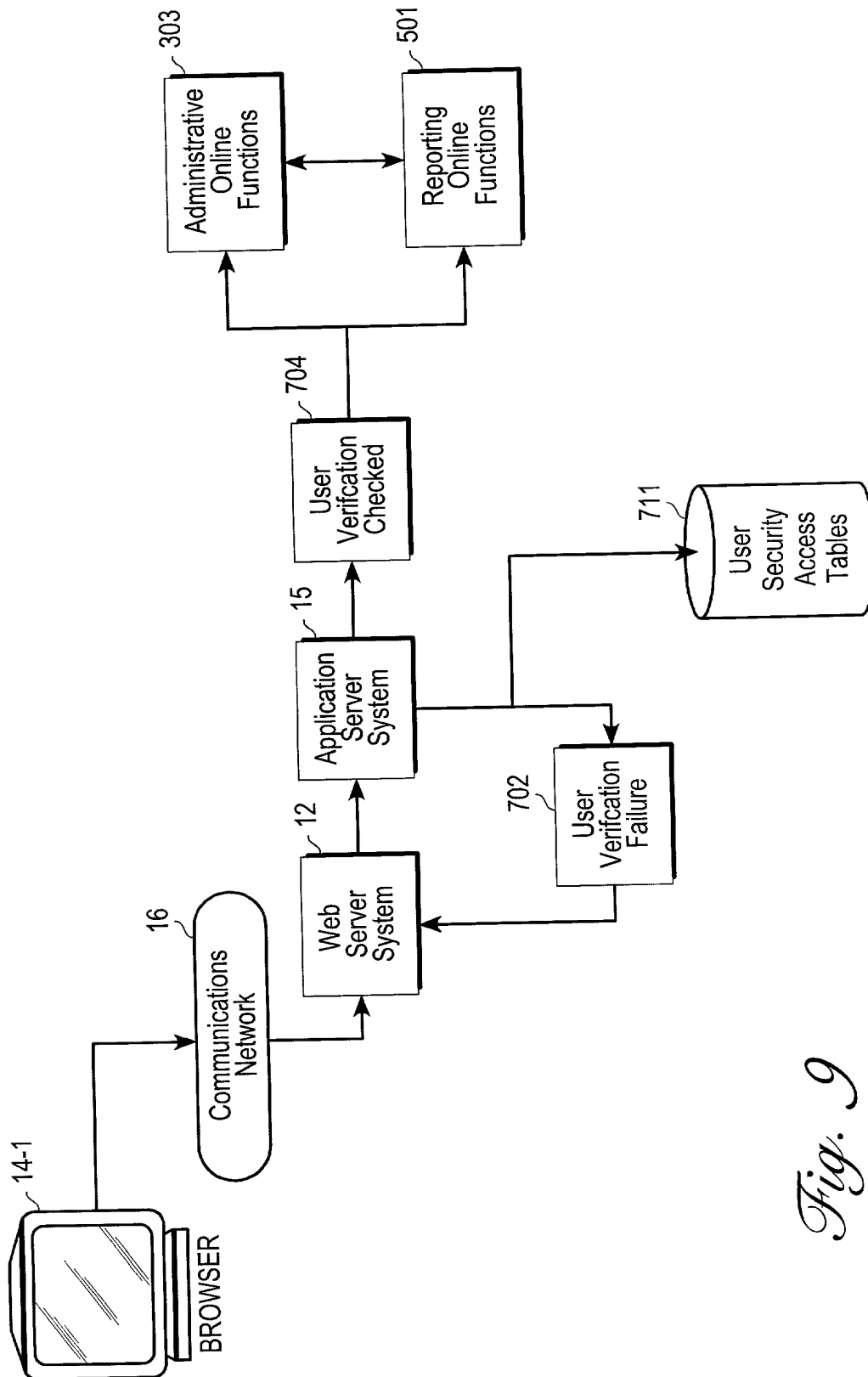
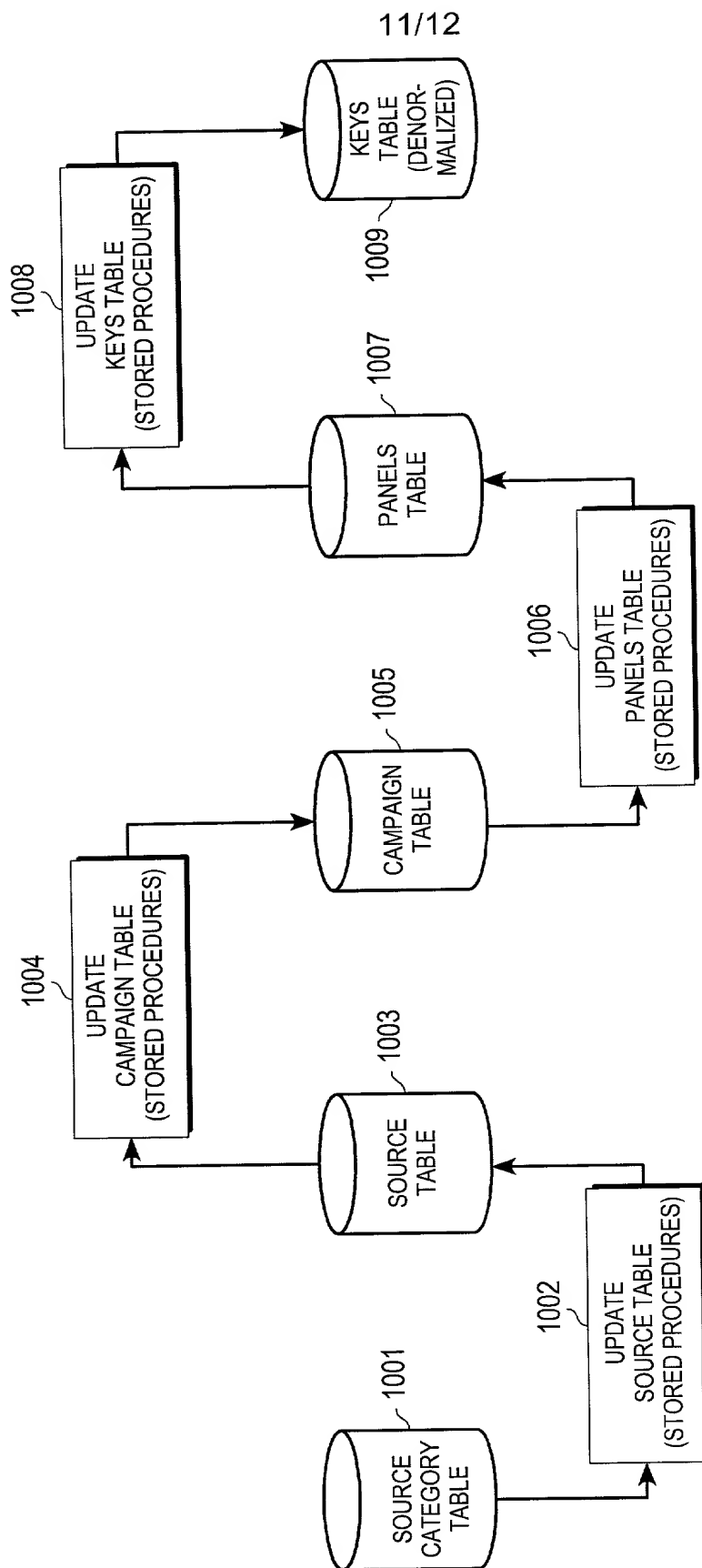


Fig. 8

*Fig. 9*

*Fig. 10*



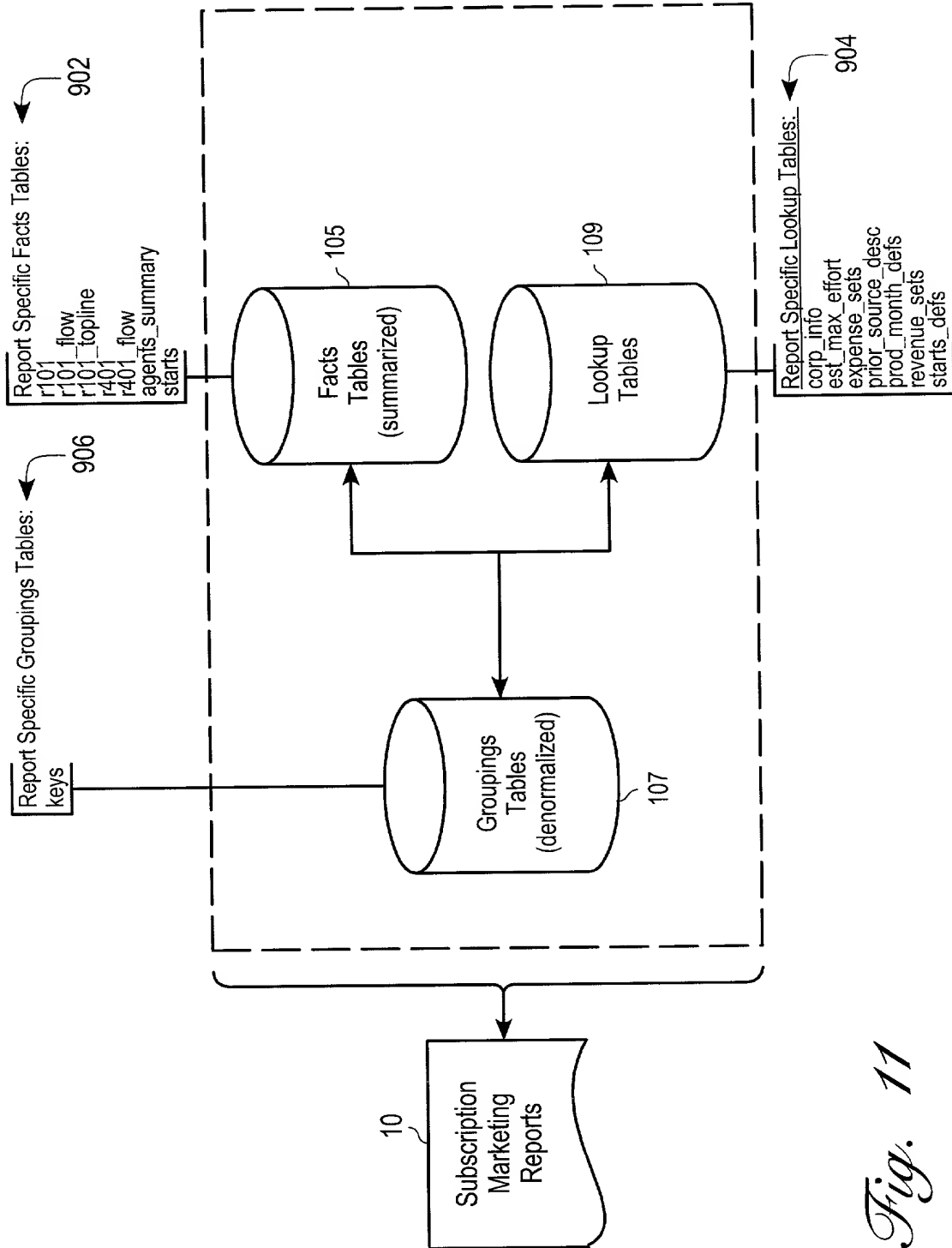
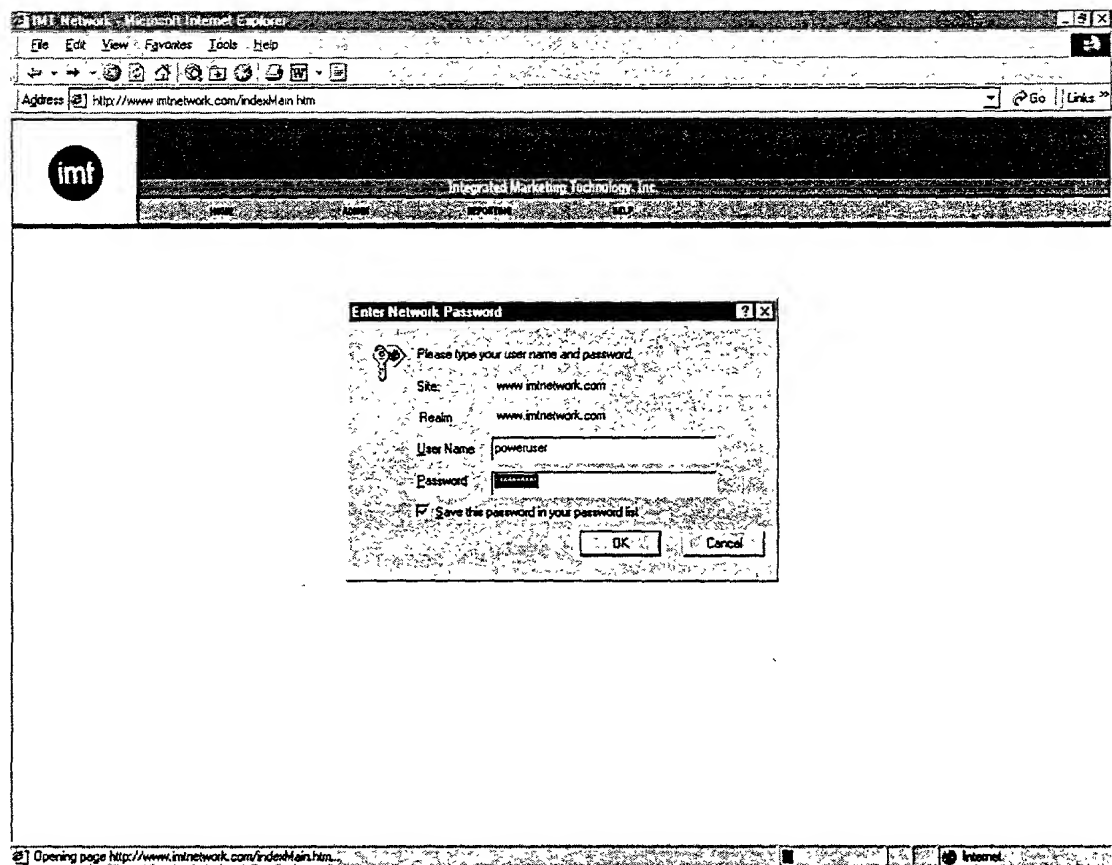


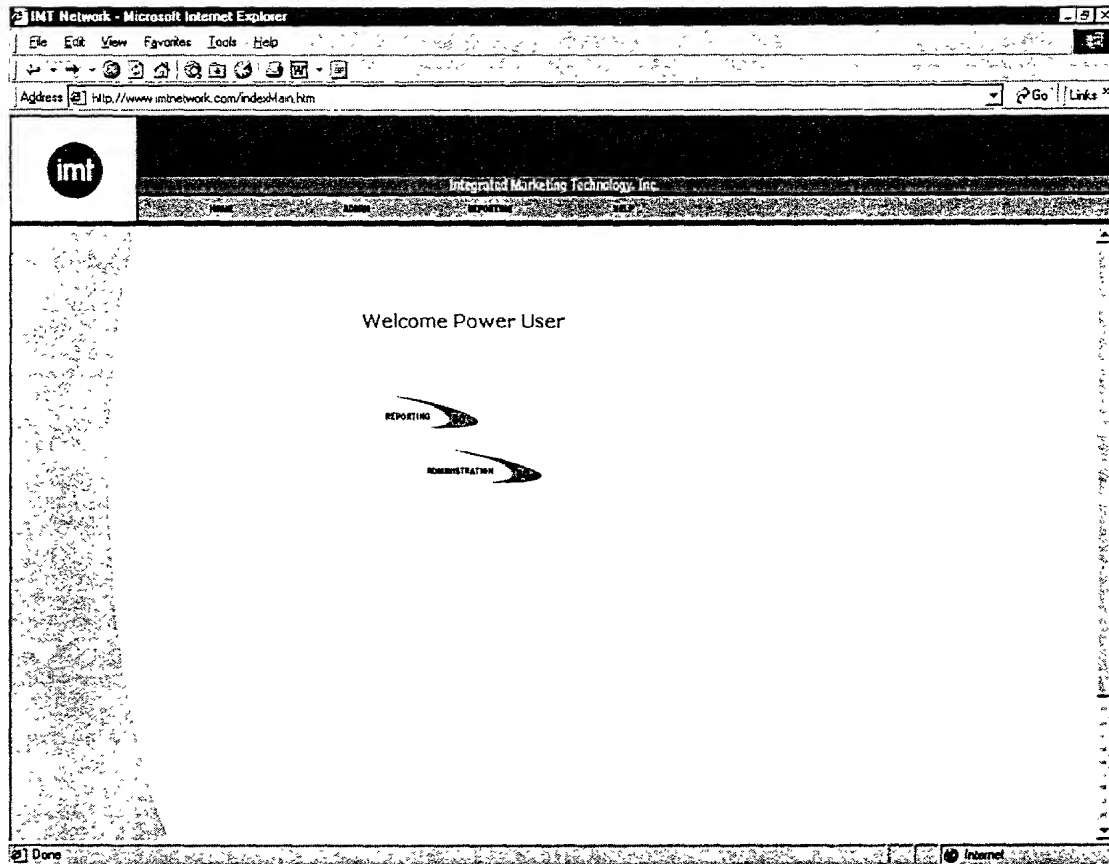
Fig. 11

Description: This page presents a user logon dialog box requiring a valid username and password in order to gain access to the application server.



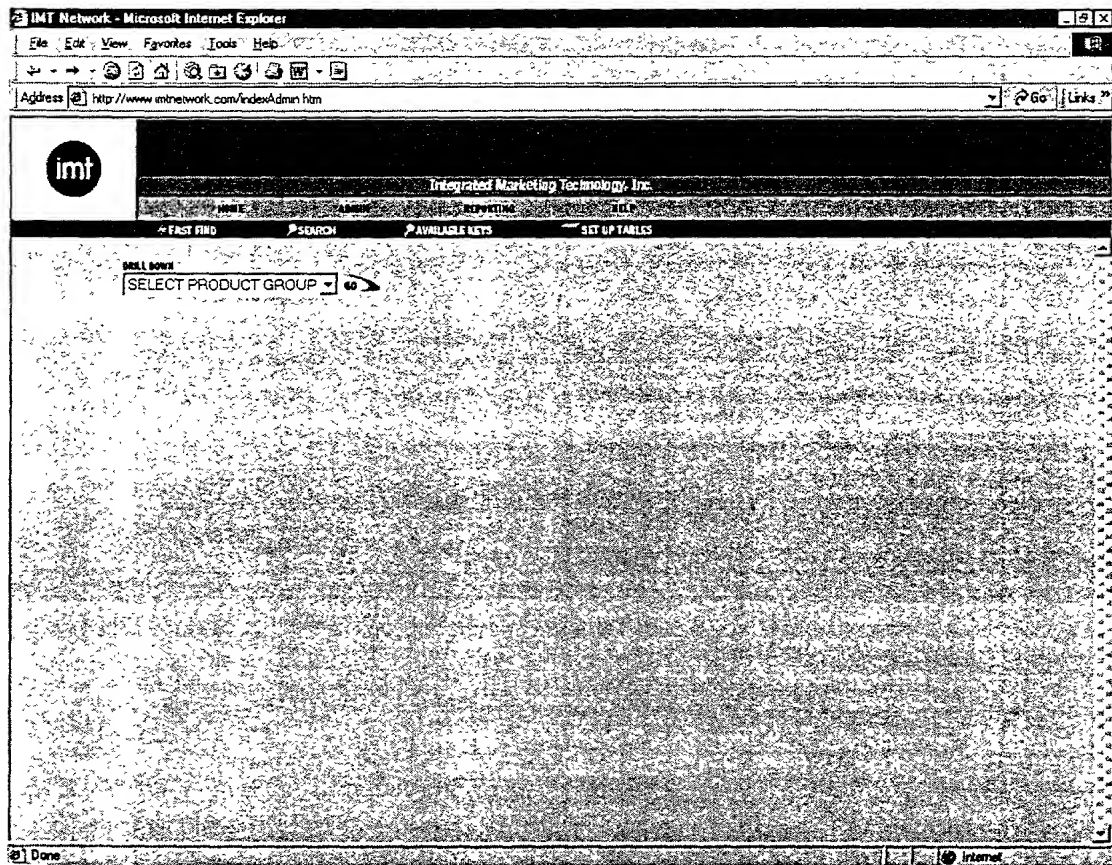
welcome.asp

Description: This page provides a user welcome message and offers an option to choose either administrative functions or reporting functions.



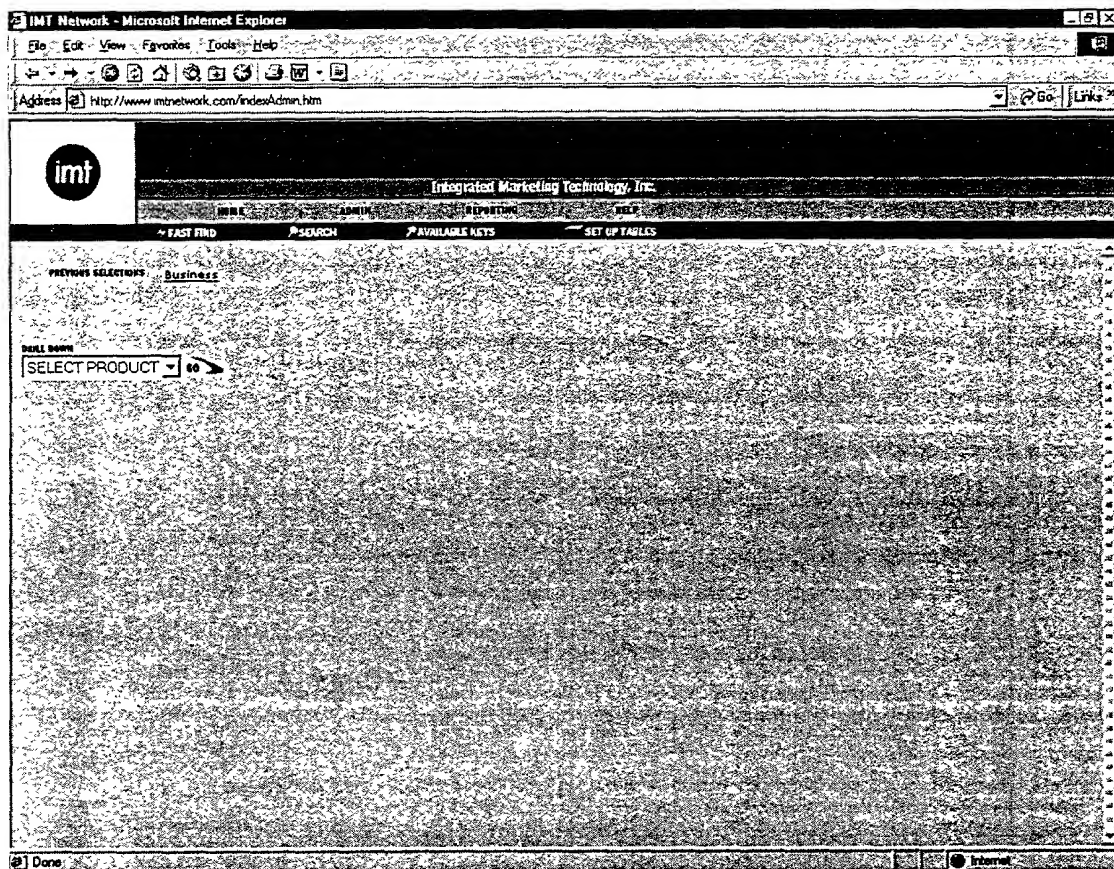
### admin1a.asp

Description: This page provides a user the ability to choose a **product group** for administration. A **product group** is defined in the **corp\_info** database table. A **product group** is a grouping in IMT's proprietary administrative hierarchy.



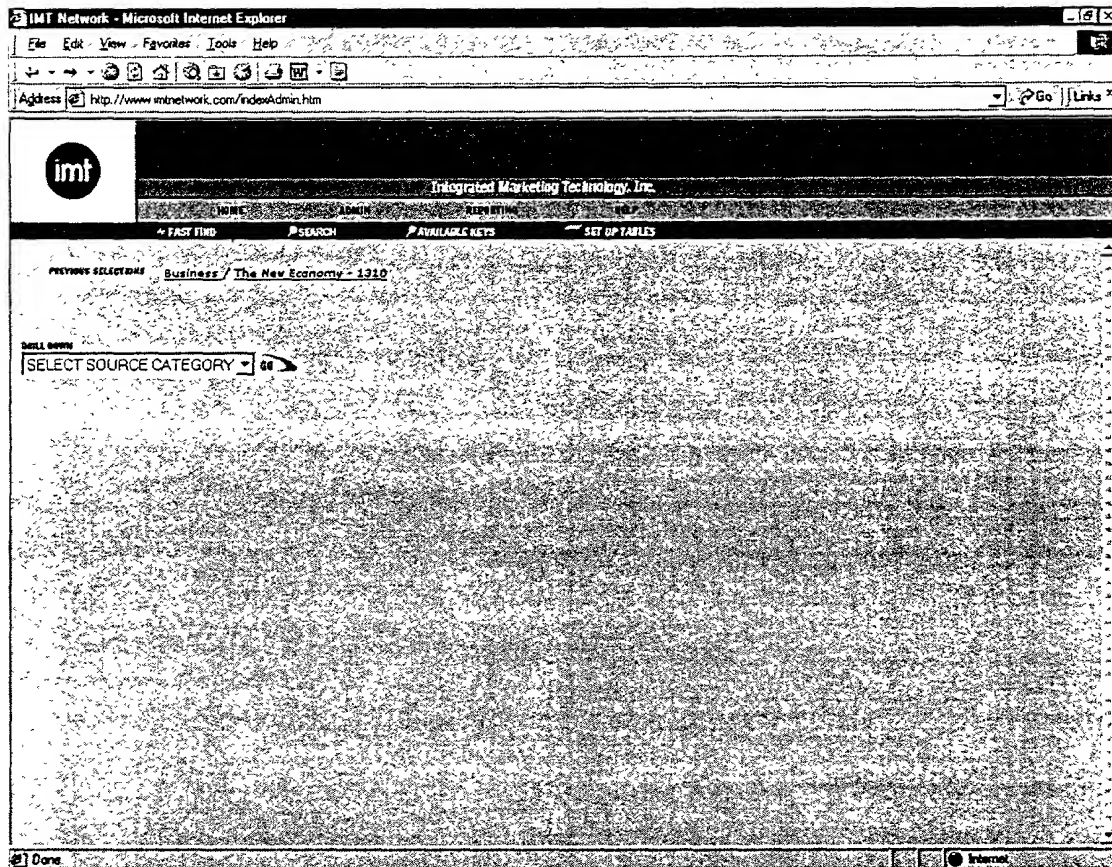
admin2a.asp

Description: This page provides a user the ability to choose a **product** for administration. A **product** is defined in the **corp\_info** database table. A **product** is a grouping in IMT's proprietary administrative hierarchy.



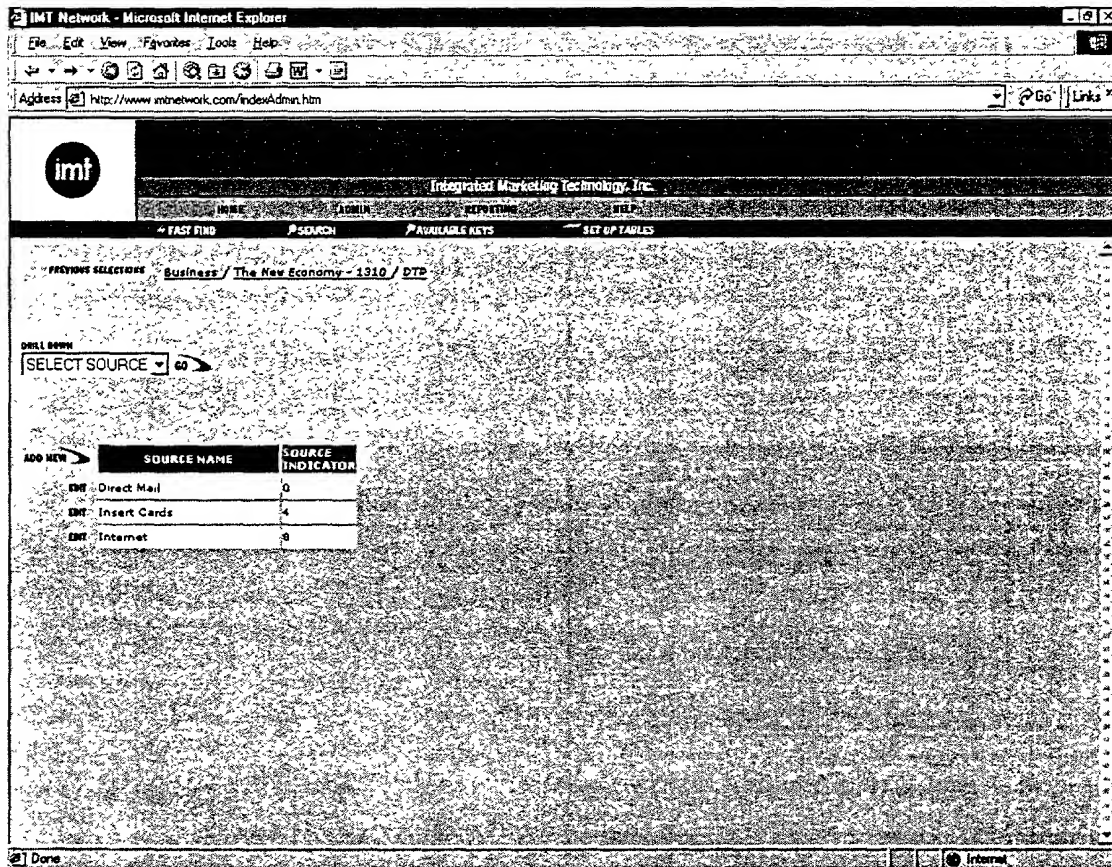
### admin3a.asp

Description: This page provides a user the ability to choose a source category for administration. A source category is defined in the `src_category` database table. A source category is a grouping in IMT's proprietary administrative hierarchy.



### admin4a.asp

Description: This page provides a user the ability to administer source records in the source database table. A source is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.





## admin5a.asp

Description: This page provides a user the ability to administer **campaign** records in the **campaigns** database table. A **campaign** is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm


**imt** Integrated Marketing Technology, Inc.

HOME ADMIN REPORTING HELP

~ FIRST FIND ~ SEARCH ~ AVAILABLE KEYS ~ SET UP TABLES

PREVIOUS SELECTIONS Business / The New Economy - 1910 / DTP / Direct Mail

ROLL DOWN

SELECT CAMPAIGN 

Show Group Edit

Check/Uncheck Add Edit Copy Delete

	CAMPAIGN NAME	CAMPAIGN DATE	CAMPAIGN TYPE	CAMPAIGN STATUS	EXPENSE SET	REVENUE SET	BUDGETED GROSS PCT	BUDGETED VOLUME
<input type="checkbox"/>	2000-06 DM	6/13/2000	2000 Summer	Active	DM 2000-06 Control	Direct Mail Test Set		
<input type="checkbox"/>	2000-03 DM	2/28/2000	2000 Spring	Active	DM 2000-03 Control	Undefined		
<input type="checkbox"/>	1999-12 DM	12/17/1999	Undefined	Closed	DM 1999-12 Control	Undefined		
<input type="checkbox"/>	1999-06 DM	6/1/1999	Undefined	Closed	DM 1999-06 Control	Undefined		

Done Internet



### admin6a.asp

Description: This page provides a user the ability to administer panel records in the panels database table. A panel is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

imt Integrated Marketing Technology, Inc.

HOME ADMIN PANELS REPORTS HELP

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

PREVIOUS SELECTIONS Business / The New Economy - 1910 / DTP / Direct Mail / 2000-06 DM

PANEL SHOW

SELECT PANEL

Show Group Edit

Check/Uncheck Add Edit Copy Delete

PANEL NAME	PANEL DATE START	PANEL TYPE	SUBTYPE	TEST TYPE	PACKAGE	EXPENSE SET	REVENUE SET
<input type="checkbox"/> 01. Continuations - 26/\$12		Continuations	Undefined	Control	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/> 02. Retests - 26/\$12		Retests	Undefined	List Test	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/> 03. List Refinements - 26/\$12		List Tests	Undefined	List Test	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/> 04. List Tests - Tier 1 - 26/\$12		List Tests	Undefined	List Test	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/> 05. List Tests - Tier 2 - 26/\$12		List Tests	Undefined	List Test	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/> 06. House Files - 26/\$12		House Files	Undefined	List Test	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/> 07. Control Test A / B - 26/\$12		Package Test	Undefined	Control Test	5 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set

Done

# adminkeyleft.asp / adminkeyright.asp

Description: This page provides a user the ability to administer key records in the keys database table. A key is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

imt Integrated Marketing Technology, Inc.

HOME ADMIN REPORTING SETUP

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

Filter Keys By Position 1 2 3 4 5 6 7 8 9 10 00

PREVIOUS SELECTIONS Business / The New Economy - 1310 / QTD / Direct Mail / 2000-06 DM / 02. Retests

26/112

Change Panel 02 Retests - 26/112

ASIGN KEYS

HIDE GROUP EDIT

SELECT ALL CLEAR ALL

AVAILABLE KEYS

<input type="checkbox"/>	0005_04/24/2000*
<input type="checkbox"/>	0019_04/24/2000*
<input type="checkbox"/>	0028_04/24/2000*
<input type="checkbox"/>	0033_04/24/2000*
<input type="checkbox"/>	0034_04/24/2000*
<input type="checkbox"/>	0046_04/24/2000*
<input type="checkbox"/>	0057_04/24/2000*
<input type="checkbox"/>	0074_04/24/2000*
<input type="checkbox"/>	0088_04/24/2000*
<input type="checkbox"/>	00AD_04/24/2000*
<input type="checkbox"/>	0107_04/24/2000*

KEY DESCRIPTION	MAIL QTY OVERRIDE / DRAW	NEWSSTAND SALE	RESPONSE BASED ON SALE (Y), DRAW (N)	LIST
(No Change)	(No Change)	(No Change)	(No Change)	(No Change)

Submit Changes

Check/Uncheck Edit Delete

PROMOTION KEY	KEY DESCRIPTION	MAIL QTY.	MAIL QTY. OVERRIDE/DRAW	NEWSSTAND SALE	RESPONSE BASED ON SALE DRAW
<input type="checkbox"/>	0BC2_20000605 thru 1/00	22002			N
<input type="checkbox"/>	0BC3_20000605 thru 1/00	29019			N

Done Internet

adminkeyleft.asp / ms\_editkey.asp

Description: This page provides a user the ability to edit key records in the keys database table. A key is a grouping in IMT's proprietary administrative hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

Go Links

imt Integrated Marketing Technology, Inc.

HOME ADMIN AVAILABLE KEYS SET UP TABLES

Filter Keys By Position

1 2 3 4 5 6 7 8 9 10 00

0

Page 1 -- Records 1-5 of 20  
Too many records selected! First 20 shown  
[Return to Menu](#)

ASSIGN KEYS

SELECT ALL CLEAR ALL

AVAILABLE KEYS

☐ 0005\_04/24/2000\*

☐ 0019\_04/24/2000\*

☐ 0028\_04/24/2000\*

☐ 0033\_04/24/2000\*

☐ 0034\_04/24/2000\*

☐ 0046\_04/24/2000\*

☐ 0057\_04/24/2000\*

☐ 0074\_04/24/2000\*

☐ 0088\_04/24/2000\*

☐ 00AD\_04/24/2000\*

☐ 0107\_04/24/2000\*

KEY	KEY DESCRIPTION	MAIL QTY	MAIL QTY OVERRIDE/DRAW	NEWSTAND SALE	RESPONSE BASED SALE OR DR (%)
OBC1_20000605		15542			N
OBCA_20000605		8886			N
OBCB_20000605		28732			N
OBC_20000605	10/99-1/00	60562			N
OBCD_20000605	4Q	11730			N

Submit Changes

Done Internet

### addsource.asp

Description: This page provides a user the ability to add source records in the source database table. A source is a grouping in IMT's proprietary administrative hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm Go Links

**imt** Integrated Marketing Technology, Inc.

HOME ADMIN REPORTING HELP

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

PREVIOUS SELECTIONS Business / The New Economy - 1310 / DTP

ADD

SUBMIT Return to Source

SOURCE NAME	SOURCE INDICATOR
SELECT SOURCE	
SELECT SOURCE	
SELECT SOURCE	
SELECT SOURCE	
SELECT SOURCE	

Done Internet

### editsource.asp

Description: This page provides a user the ability to edit source records in the source database table. A source is a grouping in IMT's proprietary administrative hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm Go Links

imt Integrated Marketing Technology, Inc.

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

PREVIOUS SELECTIONS: Business / The New Economy - 1310 / DTP

EDIT

EXPORT Return to Source

SOURCE NAME Direct Mail

SOURCE INDICATOR 0

Done Internet

## addcamp.asp

Description: This page provides a user the ability to add campaign records in the campaign database table. A campaign is a grouping in IMT's proprietary administrative hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

imt Integrated Marketing Technology, Inc.

HOME ADMIN REPORTING TOOLS

FAST FIND SEARCH AVAILABLE NETS SET UP TABLES

PREVIOUS SELECTIONS Business / The New Economy - 1310 / DTD / Direct Mail

ADD

Return to Campaign

CAMPAIGN NAME	CAMPAIGN START DATE	CAMPAIGN TYPE	CAMPAIGN STATUS	EXPENSE SET	REVENUE
		SELECT TYPE	SELECT STATUS	SELECT EXPENSE SET	SELECT REVE
		SELECT TYPE	SELECT STATUS	SELECT EXPENSE SET	SELECT REVE
		SELECT TYPE	SELECT STATUS	SELECT EXPENSE SET	SELECT REVE
		SELECT TYPE	SELECT STATUS	SELECT EXPENSE SET	SELECT REVE
		SELECT TYPE	SELECT STATUS	SELECT EXPENSE SET	SELECT REVE

Done

### ms\_editcampiagn.asp

Description: This page provides a user the ability to edit campaign records in the campaigns database table. A campaign is a grouping in IMT's proprietary administrative hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

imt Integrated Marketing Technology, Inc.

HOME ADMIN REPORTS HELP

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

Page 1 -- Records 1-4 of 4

[Return to Menu](#)

Multi Edit

CAMPAIGN NAME	CAMPAIGN START DATE	CAMPAIGN TYPE	CAMPAIGN STATUS	EXPENSE SET	REVENUE SET	
2000-06 DM	6/13/2000	2000 Summer	Active	DM 2000-06 Control	Direct Mail Test Set	
2000-03 DM	2/28/2000	2000 Spring	Active	DM 2000-03 Control	Undefined	
1999-12 DM	12/17/1999	2000 Fall	Closed	DM 1999-12 Control	Undefined	
1999-06 DM	6/1/1999	2000 Fall	Closed	DM 1999-06 Control	Undefined	

[Submit Changes](#)

Done Internet



### admin5a.asp

Description: This page provides a user the ability to administer campaign records in the campaigns database table. A **campaign** is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

**imt** Integrated Marketing Technology, Inc.

HOME ABOUT CONTACT US SET UP TABLES

FAST FIND SEARCH AVAILABLE KEYS

PREVIOUS SELECTIONS Business / The New Economy - 1310 / DTP / Direct Mail

DRILL DOWN

SELECT CAMPAIGN

Hide Group Edit

Group Edit

CAMPAIGN NAME	CAMPAIGN DATE	CAMPAIGN TYPE	CAMPAIGN STATUS	EXPENSE SET	REVENUE SET
(No Change)	(No Change)	(No Change)	(No Change)	(No Change)	(No Change)

Submit Changes

Check/Uncheck Add Edit Copy Delete

CAMPAIGN NAME	CAMPAIGN DATE	CAMPAIGN TYPE	CAMPAIGN STATUS	EXPENSE SET	REVENUE SET	BUDGETED CROSS PCT	BUDGETED VOLUME
<input type="checkbox"/> 2000-06 DM	6/13/2000	2000 Summer	Active	DM 2000-06 Control	Direct Mail Test Set		
<input type="checkbox"/> 2000-03 DM	2/28/2000	2000 Spring	Active	DM 2000-03 Control	Undefined		
<input type="checkbox"/> 1999-12 DM	12/17/1999	Undefined	Closed	DM 1999-12 Control	Undefined		

Done Internet



## admin6a.asp

Description: This page provides a user the ability to administer panel records in the panels database table. A panel is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

**imt** Integrated Marketing Technology, Inc.

HOME ADMIN REPORTING SET UP TABLES

FAST FIND SEARCH AVAILABLE ACTS

PREVIOUS SELECTIONS Business / The New Economy - 1310 / DTP / Direct Mail / 2000-06 DM

PANEL NAME  
SELECT PANEL

Hide Group Edit

Group Edit

PANEL NAME	PANEL DATE START	PANEL TYPE	SUBTYPE	TEST TYPE	
(No Change)	(No Change)	(No Change)	(No Change)	(No Change)	(No Change)

Submit Changes

Check/Uncheck Add Edit Copy Delete

	PANEL NAME	PANEL DATE START	PANEL TYPE	SUBTYPE	TEST TYPE	PACKAGE	EXPENSE SET	REVENUE SET
<input type="checkbox"/>	01. Continuations - 26/\$12		Continuations	Undefined	Control	3 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/>	02. Retests - 26/\$12		Retests	Undefined	List Test	3 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/>	03. List Refinements - 26/\$12		List Tests	Undefined	List Test	3 x 11-1/2 Silver package	DM 2000-06 Control	Direct Mail Test Set
<input type="checkbox"/>	04. List Tests - Tier 1 - 26/\$12		List Tests	Undefined	List Test	3 x 11-1/2 Silver	DM 2000-06 Control	Direct Mail Test

Done

**Description:** This page provides a user the ability to add **panel** records in the **panels** database table. A **panel** is a grouping in IMT's proprietary administrative hierarchy.

[illegible]

## ms\_editpanel.asp

Description: This page provides a user the ability to edit panel records in the panels database table. A panel is a grouping in IMT's proprietary administrative hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/indexAdmin.htm

imt Integrated Marketing Technology, Inc.

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

Page 1 -- Records 1-5 of 18  
[Return to Menu](#)

Multi Edit

PANEL NAME	PANEL DATE START	PANEL TYPE	SUBTYPE	TEST TYPE	
01 Continuations - 26/\$12		Continuations	Undefined	Control	5 x 11-1/2 Slave
02 Retests - 26/\$12		Retests	Undefined	List Test	5 x 11-1/2 Slave
03 List Refinements - 26/		List Tests	Undefined	List Test	5 x 11-1/2 Slave
04 List Tests - Tier 1 - 26/		List Tests	Undefined	List Test	5 x 11-1/2 Slave
05 List Tests - Tier 2 - 26/		List Tests	Undefined	List Test	5 x 11-1/2 Slave

Submit Changes

Done Internet

## drill.asp

Description: This page provides a user the ability to search for records in database tables using a drill-down interface. The drill-down interface corresponds to IMT's proprietary database hierarchy.

The screenshot shows a web browser window titled "http://www.imtnetwork.com/imtadmin/drill.asp - Microsoft Internet Explorer". The address bar displays "http://www.imtnetwork.com/imtadmin/drill.asp?cur". The page header features the IMT logo and the text "Integrated Marketing Technology, Inc.". Below the header is a navigation bar with links: "HOME", "ADMIN", "REPORTING", "HELP", "FAST FIND", "SEARCH", "AVAILABLE KEYS", and "SET UP TABLES". The main content area is titled "Fast Find" and displays a hierarchical tree structure. The tree starts with "Business", which expands to "The New Economy". Under "The New Economy", there are sub-items: "Agents", "Billing", "DTP", and "Direct Mail IMT". The "Direct Mail IMT" item is expanded, showing a list of 16 items, each with a number, a description, and a date/price. The items are: 01. Continuations - 26/\$12 IMT, 02. Rebate - 26/\$12 IMT, 03. List Refinements - 26/\$12 IMT, 04. List Tests - Tier 1 - 26/\$12 IMT, 05. List Tests - Tier 2 - 26/\$12 IMT, 06. House Files - 26/\$12 IMT, 07. Control Test A/B - 26/\$12 IMT, 08. Test - URL on OC & letter online subs - 26/\$12 IMT, 09. Test - Premium - 10 Principles - 26/\$12 IMT, 10. Test - Travel Alarm - 26/\$12 IMT, 11. Test - Term - 13/\$10 IMT, 12. Test - Price - 26/\$16 IMT, 13. Test - Price - 26/\$18 IMT, 14. Test - Price - 26/\$19.95 IMT, 15. Test - Price - 26/\$20 IMT, and 16. Test - Creative - New Package - 26/\$12 IMT.

Business

- 0 The New Economy
  - Agents
  - Billing
  - DTP
  - Direct Mail IMT
    - 2000-06 DM IMT
      - 01. Continuations - 26/\$12 IMT
      - 02. Rebate - 26/\$12 IMT
      - 03. List Refinements - 26/\$12 IMT
      - 04. List Tests - Tier 1 - 26/\$12 IMT
      - 05. List Tests - Tier 2 - 26/\$12 IMT
      - 06. House Files - 26/\$12 IMT
      - 07. Control Test A/B - 26/\$12 IMT
      - 08. Test - URL on OC & letter online subs - 26/\$12 IMT
      - 09. Test - Premium - 10 Principles - 26/\$12 IMT
      - 10. Test - Travel Alarm - 26/\$12 IMT
      - 11. Test - Term - 13/\$10 IMT
      - 12. Test - Price - 26/\$16 IMT
      - 13. Test - Price - 26/\$18 IMT
      - 14. Test - Price - 26/\$19.95 IMT
      - 15. Test - Price - 26/\$20 IMT
      - 16. Test - Creative - New Package - 26/\$12 IMT

## adminsearch.asp

Description: This page provides a user the ability to search for records in database tables using a combination of database selections and text string selections. Records are displayed in a table format.

Internet Explorer window: **IMT Admin Search - Microsoft Internet Explorer**  
Address: <http://www.imnetwork.com/imtadmin/AdminSearch.asp>

**imt** Integrated Marketing Technology, Inc.

~ FAST FIND ~ SEARCH ~ AVAILABLE SETS ~ SET UP TABLES

**TITLE**: The New Economy  
**SOURCE CATEGORY**: DTP  
**SOURCE**: Direct Mail  
**CAMPAIGN**: %  
**PANEL**: %

Break on: CAMPAIGN

SEARCH

	Product Group	Product Title	Source Category	Source Name	Campaign Name
<a href="#">Jump to Panels</a>	Business	The New Economy-1310	DTP	Direct Mail	1999-06 DM
<a href="#">Jump to Panels</a>	Business	The New Economy-1310	DTP	Direct Mail	1999-12 DM
<a href="#">Jump to Panels</a>	Business	The New Economy-1310	DTP	Direct Mail	2000-03 DM
<a href="#">Jump to Panels</a>	Business	The New Economy-1310	DTP	Direct Mail	2000-06 DM

Done

## avail.asp

Description: This page provides a user the ability to search for records in the available keys table using a positions of the source key record as a filtering mechanism.

Internet Explorer window: IMT Network - Microsoft Internet Explorer

Address: http://www.imtnetwork.com/mladmin/avail.asp

Integrated Marketing Technology, Inc.

FAST FIND SEARCH AVAILABLE KEYS SET UP TABLES

Filter Keys By Position

1 2 3 4 5 6 7 8 9 10 00

Product: The New Economy  
1916  
Change Product Id

SELECT ALL CLEAR ALL

AVAILABLE KEYS

<input type="checkbox"/>	0005_04/24/2000*
<input type="checkbox"/>	0019_04/24/2000*
<input type="checkbox"/>	0028_04/24/2000*
<input type="checkbox"/>	0033_04/24/2000*
<input type="checkbox"/>	0034_04/24/2000*
<input type="checkbox"/>	0046_04/24/2000*
<input type="checkbox"/>	0057_04/24/2000*
<input type="checkbox"/>	0074_04/24/2000*
<input type="checkbox"/>	0088_04/24/2000*
<input type="checkbox"/>	00AD_04/24/2000*

Done Internet



## tabautosetup.asp

Description: This page provides a user the ability to administer **autosetup** tables. Autosetup setup tables are a series of database tables that provide categorization instructions in order to build user-defined grouping records automatically. Administration is the adding, editing or deleting of records in a specific database table.

tabautosetup.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.iminetwork.com/mtadmin/tabautosetup.asp Go Links

PREVIOUS SELECTIONS Business / The New Economy - 1310 / DTP

[AutoSetup Instructions](#)

[Run Auto-Setup Wizard](#)

### AutoSetup Schemes

ADD NEW	Source	Source Indicator	Select Date From	Select Date To	Setup Mask	Expense Set
ENT	Billing	8	1/1/1998	12/31/2005	8CCxCCxxxxxxxxxP	Undefined
ENT	Renewals 1x	1	1/1/1998	12/31/2005	5xPxxZxCCC	Undefined
ENT	Renewals 2+	1	1/1/1998	12/31/2005	5xPxxZxCCC	Undefined
ENT	Agent Conversions	6	1/1/1998	12/31/2005	5xPxxZxCCC	Undefined
ENT	DTP Conversions	6	1/1/1998	12/31/2005	5xPxxZxCCC	Undefined
ENT	DTP Conversions	6	1/1/1998	12/31/2005	5xPxxZZxCCC	Undefined

### Campaign Name Definitions

ADD NEW	Source Category	Campaign Code	Campaign Description	Campaign Date	Panel Scheme
ENT	Billing	AX00	00 Credit Period AX00	8/1/2000	
ENT	Billing	VI00	00 Credit Period VI00		
ENT	Billing	0198	1998 BK 01	1/1/1998	Bill Default
ENT	Billing	0298	1998 BK 02	1/1/1998	Bill Default
ENT	Billing	0398	1998 BK 03	1/1/1998	Bill Default
ENT	Billing	0498	1998 BK 04	1/1/1998	Bill Default

### add\_autosetup.asp

Description: This page provides a user the ability to add **autosetup scheme** records in the **autosetup\_scheme** database (lookup) table. An **autosetup scheme** is a basic set of instructions for creating user defined groupings depending on the structure of a source key.

ADD

[SUBMIT](#) [Return to AutoSetup Parameters Menu](#)

SOURCE NAME

SOURCE INDICATOR

VALID FROM

VALID TO

SETUP MASK

EXPENSE SET NAME

Done Internet



### edit\_autosetup.asp

Description: This page provides a user the ability to edit **autosetup scheme** records in the **autosetup\_scheme** database (lookup) table. An **autosetup scheme** is a basic set of instructions for creating user defined groupings depending on the structure of a source key.

edit\_autosetup.asp - Microsoft Internet Explorer

Address: http://www.imtnetwork.com/mtadmin/edit\_autosetup.asp?Z=12

PREVIOUS SELECTIONS: Business / The New Economy - 1998 / DTP

EDIT

SUBMIT DELETE Return to Auto-Setup table

SOURCE NAME	Billing
SOURCE INDICATOR	8
VALID FROM	1/1/1998
VALID TO	12/31/2005
SETUP MASK	SOOxOOoooooooP
EXPENSE SET	Undefined

### add\_campaigndef.asp

Description: This page provides a user the ability to add **autosetup campaign definition** records in the **autosetup\_campaign\_def** database (lookup) table. An **autosetup campaign definition** is a basic set of instructions for creating user defined campaign groupings depending on the structure of a source key.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/add\_campaign\_def.asp

ADD

[Return to AutoSetup Parameters Menu](#)

CAMPAIGN CODE	<input type="text"/>
CAMPAIGN DESCRIPTION	<input type="text"/>
SOURCE CATEGORY	SELECT SOURCE CATEGORY
CAMPAIGN DATE	<input type="text"/>
PANEL SCHEME	SELECT PANEL SCHEME

Done Internet

### edit\_campaigndef.asp

Description: This page provides a user the ability to edit autosetup campaign definition records in the autosetup\_campaign\_def database (lookup) table. An autosetup campaign definition is a basic set of instructions for creating user defined campaign groupings depending on the structure of a source key.

edit\_autosetup\_campaign\_def.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imninet.com/mladmin/edit\_campaign\_def.asp?Z=1993 Go Links

EDIT

ADD NEW DELETE Return to Auto-Setup table

CAMPAIGN CODE	Ax00
CAMPAIGN DESCRIPTION	00 Credit Period Ax00
SOURCE CATEGORY	Billing
CAMPAIGN DATE	8/1/2000
PANEL SCHEME	SELECT PANEL SCHEME

Done Internet

### add\_renewal\_psdef.aspx

Description: This page provides a user the ability to add **autosetup prior source definition** records in the **autosetup\_renewal\_psdef** database (lookup) table. An **autosetup prior source definition** is a basic set of instructions for creating user defined source groupings depending on the structure of a source key.

ADD

[Return to AutoSetup Parameters Menu](#)

SOURCE NAME

SOURCE INDICATOR

PRIOR SOURCE INDICATOR

VALID FROM

VALID TO

Done Internet

### edit\_renewal\_psdef.asp

Description: This page provides a user the ability to edit autosetup prior source definition records in the autosetup\_renewal\_psdef database (lookup) table. An autosetup prior source definition is a basic set of instructions for creating user defined source groupings depending on the structure of a source key.

edit\_renewal\_psdef.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.iminetwork.com/enladmin/edit\_renewal\_psdef.asp?Z=2

EDIT

SOURCE NAME Agant Conversions

SOURCE INDICATOR 6

PRIOR SOURCE INDICATOR A B C D E F G H J K L M N P

VALID FROM 1/1/1999

VALID TO 12/31/2005

Return to Auto-Setup table

Done Internet



### add\_panel\_def.asp

Description: This page provides a user the ability to add **autosetup panel definition** records in the **autosetup\_panel\_def** database (lookup) table. An **autosetup panel definition** is a basic set of instructions for creating user defined panel groupings depending on the structure of a source key.

The screenshot shows a web browser window titled "IMT Network - Microsoft Internet Explorer". The address bar displays "http://www.mnnetwork.com/mtadmin/add\_panel\_def.asp". The page content includes a "Return to AutoSetup Parameters Menu" link and three input fields labeled "PANEL SCHEME", "PANEL CODES", and "PANEL NAME".

ADD

[Return to AutoSetup Parameters Menu](#)

PANEL SCHEME

PANEL CODES

PANEL NAME

Done

### edit\_panel\_def.asp

Description: This page provides a user the ability to edit **autosetup panel definition** records in the **autosetup\_panel\_def** database (lookup) table. An **autosetup panel definition** is a basic set of instructions for creating user defined panel groupings depending on the structure of a source key.

edit\_panel\_def.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mninetwork.com/mntadmin/edit\_panel\_def.asp?Z=24 Go Links

EDIT

SUBMIT DELETE Return to Auto-Setup table

PANEL SCHEME Ack Retest

PANEL CODES PZUDGJSTQCBZ1239

PANEL NAME Control

Done Internet

### setupcampstat.asp

Description: This page provides a user the ability to administer **campaign status** records in the **prod\_camp\_status** database (lookup) table. A **campaign status** is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

**Campaign Status - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/intadmin/setupCampStat.asp

**Set Up Tables**

**Campaign Status**  
The status of a Campaign (i.e. active, complete).  
Product: The New Economy-1910  
[Change Product Id](#)

[Return to Set up Table Selection Screen](#)

Campaign Status	Description
Active	
Closed	
Pending	
Undefined	

Done



Description: This page provides a user the ability to add **campaign status** records in the **prod\_camp\_status** database (lookup) table. A **campaign status** is a grouping in IMT's proprietary administrative hierarchy.

[illegible]

### editcampstat.asp

Description: This page provides a user the ability to edit **campaign status** records in the **prod\_camp\_status** database (lookup) table. A **campaign status** is a grouping in IMT's proprietary administrative hierarchy.

You are on editCampStat.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/intadmin/editCampStat.asp?Z=5

EDIT

RETURN DELETE

Return to Campaign Status Setup Table

CAMPAIGN STATUS	Active
DESCRIPTION	

Done Internet

### setupcamptype.asp

Description: This page provides a user the ability to administer campaign type records in the prod\_camp\_type database (lookup) table. A campaign type is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Campaign Type - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/setupCampType.asp Go Links

Set Up Tables

Campaign Type

A broad classification of a Campaign which is user defined and can differ between Sources. (i.e. Winter DM, hotline).

Product: The New Economy-1310

[Change Product Id](#)

[Return to Set up Table Selection Screen](#)

Campaign Type	Description
2000 Fall	
2000 Q1	
2000 Q2	
2000 Q3	
2000 Q4	
2000 Spring	
2000 Summer	
2000 Winter	
Affiliate programs	
Barter Outside Src	
Business/Free/Icono	
Foreign	
Gift	
NE web site	
Paid Outside Sources	
Snowball	
undefined	

Done Internet

### addcamptype.asp

Description: This page provides a user the ability to add **campaign type** records in the **prod\_camp\_type** database (lookup) table. A **campaign type** is a grouping in IMT's proprietary administrative hierarchy.

You are on addCampType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/imtadmin/addCampType.asp Go Links

PREVIOUS SELECTIONS: Business / The New Economy - 1310

ADD

SUBMIT Return to Campaign Type Setup Table

CAMPAIGN TYPE	DESCRIPTION

Done Internet

### editcamptype.asp

Description: This page provides a user the ability to edit **campaign type** records in the **prod\_camp\_type** database (lookup) table. A **campaign type** is a grouping in IMT's proprietary administrative hierarchy.

You are on editCampType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/mtadmin/editCampType.asp?Z=17 Go Links

EDIT

RETURN DELETE

Return to Campaign Type Setup Table

CAMPAIGN TYPE NAME	2000 Fall
DESCRIPTION	

Done Internet

### setupeset.asp

Description: This page provides a user the ability to administer **expense set** records in the **expense\_sets** database (lookup) table. Administration is the adding, editing or deleting of records in a specific database table.

**Expense Set - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/mtnadmin/setupESet.asp Go Links

**Set Up Tables**  
**Expense Set**  
A set of expenses which can be applied to a Campaign, Panel or Key.  
Product: The New Economy-1318  
[Change Product Id](#)

[Return to Set up Table Selection Screen](#)

Expense Set
DM 1999-06 Control
DM 1999-06 No Broch Test
DM 1999-06 Premium Test
DM 1999-06 Prof Rate Test
DM 1999-12 Control
DM 1999-12 Premium Test
DM 1999-12 Prof Rate Test
DM 1999-12 Prof Rate/Tax RptTest
DM 1999-12 Sticker Test
DM 1999-12 Welcome Pkg (6x9) Test
DM 2000-03 2 Free Issues
DM 2000-03 3 Free Issues
DM 2000-03 Control
DM 2000-03 Premium Test
DM 2000-06 Control
DM 2000-06 Premium Test
Generic Renewals
Inserts - Bindin - Control
Inserts - Blowin - Control
Inserts - dbl Bindin - Nov Dec99 test
Undefined

Done Internet



### addeset.asp

Description: This page provides a user the ability to add expense set records in the expense\_sets database (lookup) table.

ADD

[Return to Expense Set Setup Table](#)

Expense Set	<input type="text"/>	CPM	Mail Qty
Print Cost	<input type="text"/>	CPM	Mail Qty
Merge Purge Cost	<input type="text"/>	CPM	Mail Qty
Lettershop Cost	<input type="text"/>	CPM	Mail Qty
Postage Out Cost	<input type="text"/>	CPM	Mail Qty
Postage In Cost	<input type="text"/>	CPM	Gross Subs
Premium Cost	<input type="text"/>	CPM	Gross Subs
Bad Pay Cost	<input type="text"/>	CPM	Gross-Net Subs
Billing Cost	<input type="text"/>	CPM	Credit Subs
Sub Service Cost	<input type="text"/>	CPM	Net Subs
Exp Other Cost	<input type="text"/>	CPM	Mail Qty

Done

### editset.asp

Description: This page provides a user the ability to edit expense set records in the expense\_sets database (lookup) table.

editset.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/intadmin/editset.asp?Z=39

Return to Expense Set Setup Table

Expense Set	DM 1999-06 Control		
Print Cost	135	CPM	Mail Qty
Merge Purge Cost	12	CPM	Mail Qty
Lettershop Cost	29.55	CPM	Mail Qty
Postage Out Cost	0.165	CPU	Mail Qty
Postage In Cost	0.34	CPU	Gross Subs
Premium Cost		CPU	Gross Subs
Bad Pay Cost	2.26	CPU	Gross-Net Subs
Billing Cost	1.2	CPU	Credit Subs
Sub Service Cost		CPM	Net Subs
Exp Other Cost		CPM	Mail Qty

Done



### setuplistcat.asp

Description: This page provides a user the ability to administer list category records in the list\_cats database (lookup) table. A list category is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Set Up Tables  
List Category  
A broad category of related lists.  
Product: The New Economy-1310  
Change Product Id

[Return to Set up Table Selection Screen](#)

List Category	Description
Business Forums (seminar attendees)	
Business Mag	
Computer Entertainment	
Computer Software Journals (paid)	
Hi-Tech Newsletter (paid)	
Journal of MIT Sloan School of Mgmt (paid)	
Mag (controlled)	
Pub (paid)	
Undefined	

### addlistcat.asp

Description: This page provides a user the ability to add list category records in the list\_cats database (lookup) table. A list category is a grouping in IMT's proprietary administrative hierarchy.

You are on addlistCat.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/imtadmin/addlistCat.asp Go Links

ADD

[Return to List Category Setup Table](#)

LIST CATEGORY	DESCRIPTION

Done

Start Gregg Wilkin... Technical Do... ASP PAGES.d... Web Screen... [02] 01:43 - IMT Network... You are on... 11:22 AM

### editlistcat.asp

Description: This page provides a user the ability to edit list category records in the list\_cats database (lookup) table. A list category is a grouping in IMT's proprietary administrative hierarchy.

You are on editListCat.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/imtadmin/editListCat.asp?Z=16 Go Links

**EDIT**

**SAVE** **DELETE**

[Return to List Category Setup Table](#)

<b>LIST CATEGORY</b>	Business Mag
<b>DESCRIPTION</b>	

Done

Start Gregg Wilkin... Technical D... ASP PAGES.d... Web Screen... [02] 02-25 - C... IMT Network... You are on... 11:23 AM

### setuplistname.asp

Description: This page provides a user the ability to administer list name records in the list\_names database (lookup) table. A list name is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Set Up Tables  
List Name  
The name of a list.  
Product: The New Economy-  
1310  
[Change Product Id](#)

[Return to Set up Table  
Selection Screen](#)

List Name	Description
Entrepreneur Mag	
Extra Income	
Fast Company	
Forbes	
Forbes Investment Newsletter	
Forbes Investment Newsletter (formerly Forbes Speciality Publication)	
Forbes Magazine	
Forbes Speciality Publication	
Forbes Speciality	
Fortune	
Gilder Tech Report/Forbes Investment Weekly	
Gizmo	
Global Finance Mag	
Global Finance Magazine	
Global Insights	
Harvard Business Review	
Harvard Management Update	
Harvard Mgmt Update	
Hume Group / Supennvestor Files	
Hume Group The Superinvestor File	
Hume Group The Superinvestor Files	
INC Magazine	
Inc. Mag	
Inc. Magazine	

### addlistname.asp

Description: This page provides a user the ability to add list name records in the list\_names database (lookup) table. A list name is a grouping in IMT's proprietary administrative hierarchy.

You are on addListName.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadminv/addListName.asp

ADD

Return to List Name Setup Table

LIST NAME	DESCRIPTION

Done Internet

### editlistname.asp

Description: This page provides a user the ability to edit list name records in the list\_names database (lookup) table. A list name is a grouping in IMT's proprietary administrative hierarchy.

You are on editListName.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/enladminv/editListName.asp?Z=297

EDIT

RETURN DELETE

Return to List Name Setup Table

LIST NAME	Entrepreneur Mag
DESCRIPTION	

Done Internet



### setuplistsegment.asp

Description: This page provides a user the ability to administer list segments records in the list\_segments database (lookup) table. A list segment is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Set Up Tables  
List Segment  
A specific selection from a list. (i.e. 90 day purchasers, expires, male).  
Products: The New Economy-1319  
[Change Product Id](#)

[Return to Set up Table Selection Screen](#)

List Segment	Description
expire names thru 2/00	
Paid Expires	
Paid Expires (3 mos)	
1/99 - 2/99 Active Paid Subscribers	
1/99 - 3/99 active paid subs	
1/99 - 3/99 active subs	
1/99 - 3/99 active US paid subs	
1/99 - 3/99 subs (7,000 names on exchange)	
1/99 Active Paid Subscribers	
1/99 Active Subs	
1/99 active US subs	
1/99-5/99 US Active Paid Subscribers	
1/99-6/99 Hotline Active Members	
10/99 - 1/00 active paid subs	
10/99 - 11/99 Active paid subs	
10/99-2/00 Paid Active DTP Subs	
10/99-2/00 US Active Paid Subs/male	
11/98 - 1/99 buyers	
11/99 Active paid subs (7,300 on exchange)	
11/99 paid subs	
11/99-2/00 active paid subs	
11/99-2/00 active paid subs (omit concurrent)	
11/99-2/00 Active paid subs/male/173Kincome	
11/99-2/00 Active subs	

### addlistsegments.asp

Description: This page provides a user the ability to add list segments records in the list\_segments database (lookup) table. A list segment is a grouping in IMT's proprietary administrative hierarchy.

You are on addListSegment.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/addListSegment.asp

ADD

[Return to List Segment Setup Table](#)

LIST SEGMENT	DESCRIPTION

Done



### editlistsegments.asp

Description: This page provides a user the ability to edit **list segments** records in the **list\_segments** database (lookup) table. A **list segment** is a grouping in IMT's proprietary administrative hierarchy.

You are on editListSegment.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/entadmin/editListSegment.asp?Z=155

EDIT

SUBMIT DELETE

Return to List Segment Setup Table

LIST SEGMENT	Paid Expires
DESCRIPTION	

Done Internet

### setuppanelpackage.asp

Description: This page provides a user the ability to administer panel package records in the prod\_panel\_package database (lookup) table. A panel package is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Panel Package - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/entadmin/setupPanelPackage.asp

Set Up Tables

Panel Package  
The code or description of a specific mailed package.

Product: The New Economy-1910

[Change Product Id](#)

[Return to Set up Table Selection Screen](#)

Panel Package	Description
#10 package	
5 x 11-1/2 Silver package	
5 x 11-1/2 Silver package - travel alarm	
5 x 11-1/2 Silver package /10 Prindples	
6 x 9 package	
6 x 9 package - mousepad	
6 x 9 package - travel alarm	
Form 1	Internet
Form 2	Internet
Form 3	Internet
Form 4	Internet
Form 5	Internet
Saf-t	Internet
undefined	undefined

Done Internet

### addpanelpackage.asp

Description: This page provides a user the ability to add panel package records in the prod\_panel\_package database (lookup) table. A panel package is a grouping in IMT's proprietary administrative hierarchy.

You are on addPanelPackage.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/intadmin/addPanelPackage.asp

Go Links

ADD

SUBMIT Return to Panel Package Setup Table

PANEL PACKAGE	DESCRIPTION

Done Internet

### editpanelpackage.asp

Description: This page provides a user the ability to edit panel package records in the prod\_panel\_package database (lookup) table. A panel package is a grouping in IMT's proprietary administrative hierarchy.

You are on editPanelPackage.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/editPanelPackage.asp?Z=11

EDIT

SUBMIT DELETE

Return to Panel Package Setup Table

PANEL PACKAGE	#10 package
DESCRIPTION	

Done Internet

### setuppanelsubtype.asp

Description: This page provides a user the ability to administer panel subtype records in the **corp\_panel\_sub\_type** database (lookup) table. A **panel subtype** is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

**Set Up Tables**  
Panel Sub Type  
Another field used to further define a set of related Panels (if necessary). (i.e. sweepstakes, newstand, inserts).

[Return to Set-up Table Selection Screen](#)

Panel Sub Type	Description
Bind-in (Double)	Insert Cards
Bind-in (Single)	Insert Cards
Bind-in (Triple)	Insert Cards
Blow-in	Insert Cards
Hard Offer	Internet
I - Banner	Internet
I - Bill-Me	Internet
I - Button	Internet
I - E-Mail	Internet
I - Free Trial	Internet
I - Gift	Internet
I - Hard Offer	Internet
I - Interstitial	Internet
I - Payment Option	Internet
I - Renewal	Internet
I - Text Link	Internet
Soft Offer	Internet
Undefined	Undefined

### addpanelsubtype.asp

Description: This page provides a user the ability to add **panel subtype** records in the **corp\_panel\_sub\_type** database (lookup) table. A **panel subtype** is a grouping in IMT's proprietary administrative hierarchy.

You are on addPanelSubType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mladmin/addPanelSubType.asp

ADD

Submit Return to Panel Sub Type Setup Table

PANEL SUB TYPE	DESCRIPTION

Done Internet



### editpanelsubtype.asp

Description: This page provides a user the ability to edit **panel subtype** records in the **corp\_panel\_sub\_type** database (lookup) table. A **panel subtype** is a grouping in IMT's proprietary administrative hierarchy.

You are on editPanelSubType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/editPanelSubType.asp?Z=77

EDIT

RETURN DELETE

Return to Panel Sub Type Setup Table

PANEL SUB TYPE	Hard Offer
DESCRIPTION	Internet

Done Internet

### setuppaneltesttype.asp

Description: This page provides a user the ability to administer panel test type records in the corp\_panel\_test\_type database (lookup) table. A panel test type is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Panel Sub Type - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.imtnetwork.com/mtadmin/setupPanelTestType.asp> Go Links

**Set Up Tables**  
**Panel Test Type**  
A field used to define a set of related panels by test method (i.e. price, offer, term).

[Return to Set up Table Selection Screen](#)

Panel Test Type	Description
Backtest (old control)	
Billing Test	Billing Test
Control	
Control Test	DM
Copy Test	DM
Creative Test	DM
Creative Test: C. Eler	Covers vs. Cool Free Stuff
Creative Test: Savings Chart	Lara Croft vs. Savings Chart
Creative Test: Single Bind-In	Triple vs. Single Bind-In
E-Mail Test	no e-mail vs. e-mail optional
Extra Blow-in Test	
Foreign w/CD	Internet
Holiday Gift Card Test	response cannibalization test
I - Form Test	Internet
I - Offer Test	Internet
I - Price Test	Internet
List Test	DM
No Premium	
Offer Test	DM
Package Test	DM
Premium Test	
Premium Test: Best of CD	EPC2 vs. EPC2/BCD
Premium Test: Everglide Mouse Pad	EPC3/SG2 vs. Everglide Mouse Pad
Premium Test: Reviews Index	EPC2/SG1 vs. EPC2/R1
Premium Test: Strategy Guide	EPC2 vs. EPC2/SG1

Done Internet



### addpaneltesttype.asp

Description: This page provides a user the ability to add panel test type records in the corp\_panel\_test\_type database (lookup) table. A panel test type is a grouping in IMT's proprietary administrative hierarchy.

You are on addPanelTestType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/mtadmin/addPanelTestType.asp

ADD

SUBMIT Return to Panel Test Type Setup Table

PANEL TEST TYPE	DESCRIPTION

Internet

### editpaneltesttype.asp

Description: This page provides a user the ability to edit panel test type records in the corp\_panel\_test\_type database (lookup) table. A panel test type is a grouping in IMT's proprietary administrative hierarchy.

You are on editPanelTestType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/editPanelTestType.asp?Z=4

EDIT

SUBMIT DELETE

Return to Panel Test Type Setup Table

PANEL TEST TYPE	Control
DESCRIPTION	

Done Internet

### setuppaneltype.asp

Description: This page provides a user the ability to administer panel type records in the corp\_panel\_type database (lookup) table. A panel type is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Panel Type - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/setupPanelType.asp

Set Up Tables

Panel Type  
A set of related panels typically used to distinguish control panels from test panels.

[Return to Set up Table Selection Screen](#)

Panel Type	Description
Continuations	
Control	Control
Form 1	
Form 2	
Gift	Gift
House Files	
I - 3Dgaming.com	Internet
I - All In One	Internet
I - Alloy	Internet
I - Asktherobot	Internet
I - 828 Free Network	Internet
I - Banner	Internet
I - Company Sleuth	Internet
I - Customer Service	Internet
I - Doomedevilator	Internet
I - E-Mail Bill	Internet
I - E-Mail Offer	Internet
I - E-Mail Renewal	Internet
I - EB	Internet
I - Entry Interstitial	Internet
I - Epitonic	Internet
I - Exit Interstitial	Internet
I - FGM Online	Internet
I - Foreign Address Forms	
I - Free Offer Link	Internet

Done Internet

### addpaneltype.asp

Description: This page provides a user the ability to add **panel type** records in the **corp\_panel\_type** database (lookup) table. A **panel type** is a grouping in IMT's proprietary administrative hierarchy.

You are on addPanelType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/imtadmin/addPanelType.asp

ADD

[Return to Panel Type Setup Table](#)

PANEL TYPE	DESCRIPTION

Done Internet

### editpaneltype.asp

Description: This page provides a user the ability to edit panel type records in the corp\_panel\_type database (lookup) table. A panel type is a grouping in IMT's proprietary administrative hierarchy.

You are on editPanelType.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/mtnadmin/editPanelType.asp?Z=3

Go Links

EDIT

SUBMIT DELETE

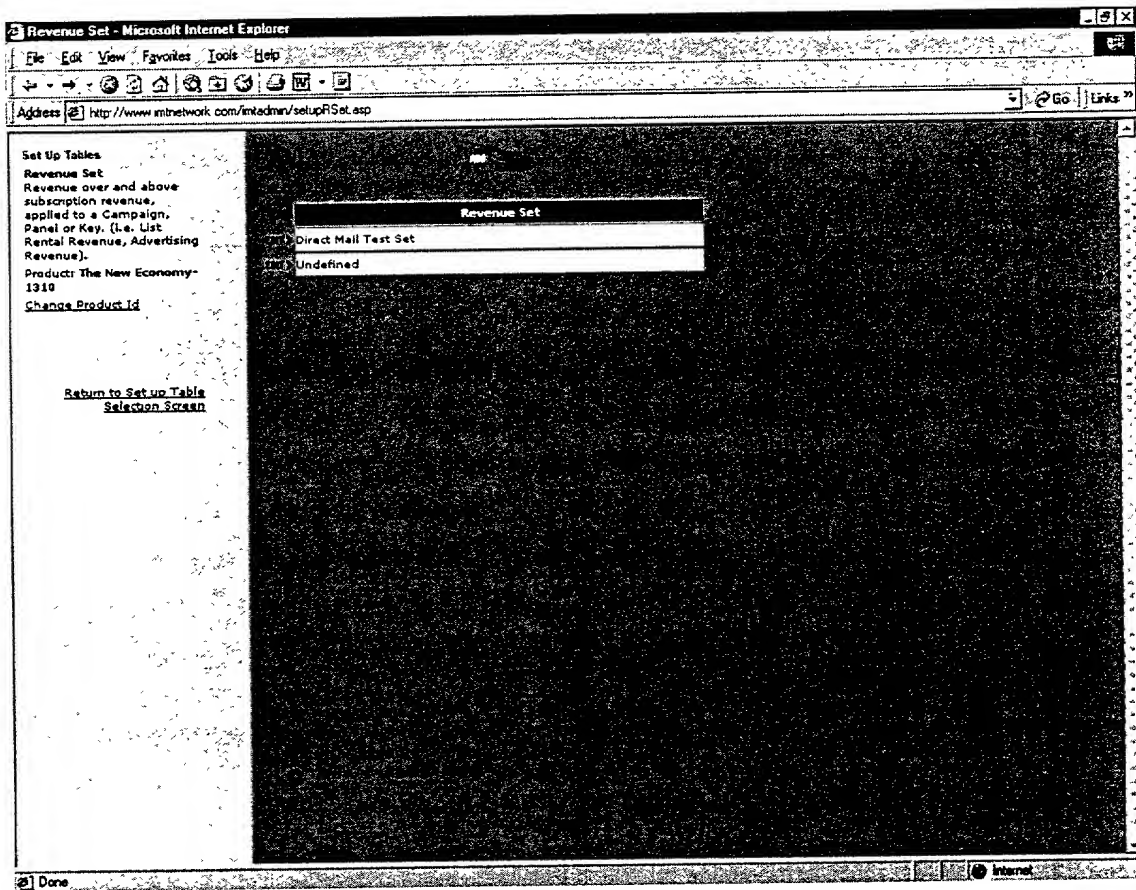
Return to Panel Type Setup Table

PANEL TYPE	Continuations
DESCRIPTION	

Done Internet

### setuprset.asp

Description: This page provides a user the ability to administer **revenue set** records in the **revenue\_sets** database (lookup) table. Administration is the adding, editing or deleting of records in a specific database table.





### addrset.asp

Description: This page provides a user the ability to add revenue set records in the revenue\_sets database (lookup) table.

addrset.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.minetwork.com/mtadmin/addrset.asp

Go Links

ADD

SUBMIT

Return to Revenue Set Setup Table

Revenue Set	<input type="text"/>	
Sub Revenue	<input type="text"/>	CPM or CPU
Ad Revenue	<input type="text"/>	CPM or CPU
List Rental Revenue	<input type="text"/>	CPM or CPU
Other Revenue	<input type="text"/>	CPM or CPU

Start Gregg Williams - Inbox ... Technical Documents IMT Network - Micros... Web Screen Shots.doc addrset.asp - Micr... 3:07 PM

### editrset.asp

Description: This page provides a user the ability to edit revenue set records in the revenue\_sets database (lookup) table.

editrset.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imnetwork.com/imadmin/editrset.asp?Z=5 Go Links

EDIT

Return to Revenue Set Setup Table

Revenue Set	Direct Mail Test Set	
Sub Revenue		CPM or CPU
Ad Revenue		CPM or CPU
List Rental Revenue	1	CPU
Other Revenue		CPM or CPU

Done Internet



### setupsourcename.asp

Description: This page provides a user the ability to administer **source description** records in the **source\_descriptions** database (lookup) table. A **source description** is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Source Description - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/setupSourceName.asp

Set up Tables  
Source Description  
Describes a specific channel  
of sale.

[Return to Set up Table  
Selection Screen](#)

Source	Description
Agent Conversions	Renewals
Auto Renewals	Renewals
Billing	Billing
Cold Donor	Gift
Direct Mail	DTP
Donor Renewal	Gift
DTP Conversions	Renewals
Gift Insert Cards	Gift
In Box/Card Decks	7
Insert Cards	DTP
Internet	DTP
Internet - Banner	DTP - Internet
Internet - Link	DTP - Internet
Internet - Naviant	DTP
Internet - Opt-In	DTP - Internet
Internet Gift	Gift
Renewals 1x	Renewals
Renewals 2+	Renewals
Space Ads	5

Done Internet

### addsourceDesc.asp

Description: This page provides a user the ability to add **source description** records in the **source\_descriptions** database (lookup) table. A **source description** is a grouping in IMT's proprietary administrative hierarchy.

You are on addSourceName.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.mtnetwork.com/mtadmin/addSourceName.asp

ADD

[Return to Source Description Setup Table](#)

SOURCE	DESCRIPTION

Done

### edit sourcedesc.asp

Description: This page provides a user the ability to edit source description records in the source\_descriptions database (lookup) table. A source description is a grouping in IMT's proprietary administrative hierarchy.

You are on editSourceName.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/mtadmin/editSourceName.asp?Z=1

EDIT

SOURCE DESCRIPTION

SOURCE Direct Mail

DESCRIPTION DTP

Return to Source Description Setup Table

Done Internet

### setupprior sourcedesc.asp

Description: This page provides a user the ability to administer **source identifier description** records in the **prior\_source\_desc** database (lookup) table. A **source identifier description** is a grouping in IMT's proprietary administrative hierarchy. Administration is the adding, editing or deleting of records in a specific database table.

Prior Source Description - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/imtadmin/setupPriorSrcDesc.asp

Set Up Tables  
Source Identifier  
Describes Prior Sources  
Product: The New Economy - 1310  
Change Product Id

[Return to Set up Table Selection Screen](#)

Source Identifier	Renewal Prior Source Description	Billing Current Source Description	Model Source Category	Model Source Description	Agent Source Description
INT 0	Direct Mail	Direct Mail	DTP		0
INT 1	Renewals	Renewals	Renewals		1
INT 2	Gift	Gift	Gifts		2
INT 3	White Mail	White Mail	DTP		3
INT 4	Inserts	Insert Cards	DTP		4
INT 5	Space Ads	Space Ads	DTP		5
INT 5B	Misc DTP	Misc DTP	DTP	DTP	
INT 5G	Group Subs	Group Subs	DTP	5G	5G
INT 5S	Sponsored Subs	Sponsored Subs	DTP	5S	5S
INT 6	Conversions	Conversions	Renewals		F
INT 7	In Box Promotions	In Box Promotions	DTP		7
INT 8	Internet	Internet	DTP		8
INT 9	Control to Paid	Ctl. Conv. to Paid	DTP		9
INT A	PCH	N/A	Agents		PCH
INT B	Cash Field	N/A	Agents		Cash Field
INT C	PDS	N/A	Agents		PDS
INT D	Direct Mail Agts	N/A	Agents		Direct Mail Agts
INT E	DMI Sponsor	N/A	Agents		DMI Sponsor
INT F	F	F	Agents		F
INT G	School Sponsored	N/A	Agents		School Sponsored
INT H	Catalogue	N/A	Agents		Catalogue
INT I	I	I	Agents		I
INT J	J	J	Agents		J

Done Internet

Description: This page provides a user the ability to add **source identifier description** records in the **prior\_source\_desc** database (lookup) table. A **source identifier description** is a grouping in IMT's proprietary administrative hierarchy.

[illegible]

### editprior sourcedesc.asp

Description: This page provides a user the ability to edit source identifier description records in the prior\_source\_desc database (lookup) table. A source identifier description is a grouping in IMT's proprietary administrative hierarchy.

You are on editPriorSrcDesc.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/imtadmin/editPriorSrcDesc.asp?Z=73

EDIT

SUBMIT DELETE

Return to Prior Source Description Setup Table

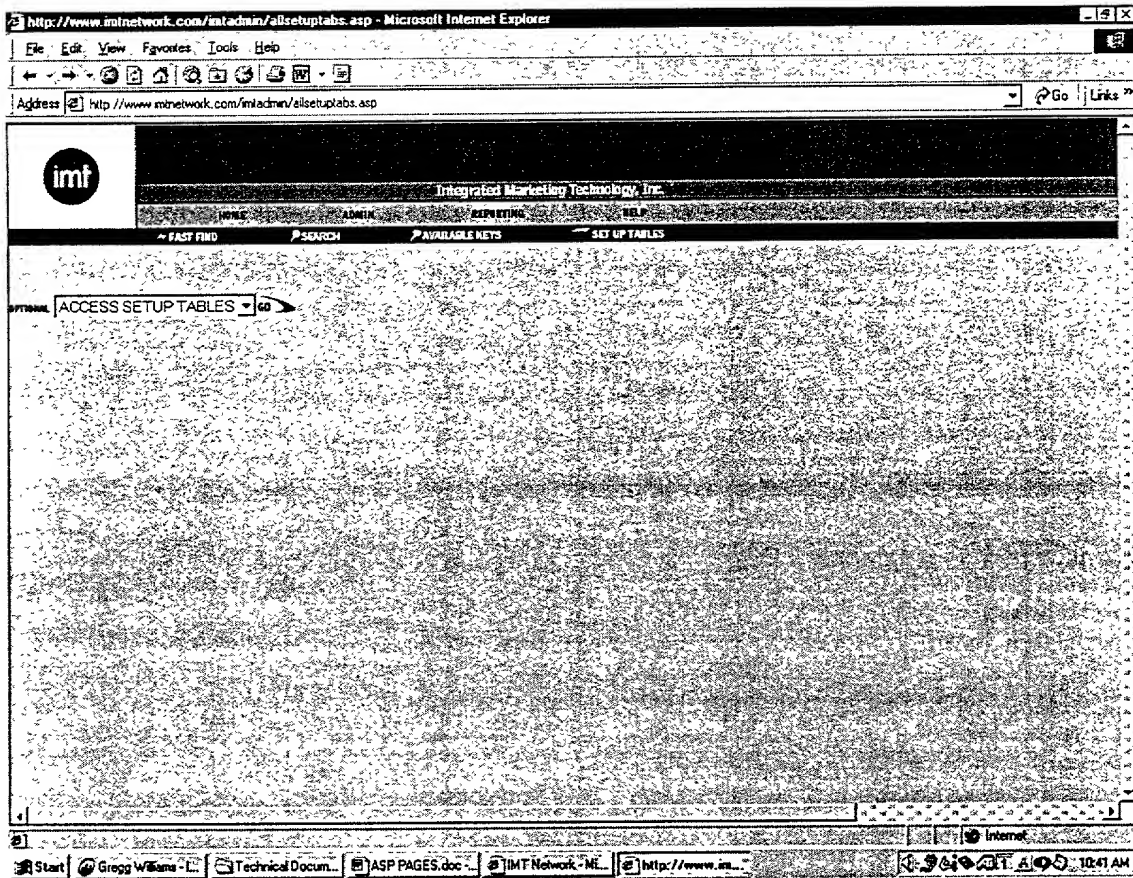
PRIOR SOURCE IDENTIFIER	0
RENEWAL PRIOR SOURCE DESC	Direct Mail
BILLINGS CURRENT SOURCE DESC	Direct Mail
MODEL SOURCE CATEGORY	DTP
MODEL SOURCE DESCRIPTION	
AGENT SOURCE DESCRIPTION	0
VALID FROM DATE	1/1/1996
VALID TO DATE	1/1/2010

Done

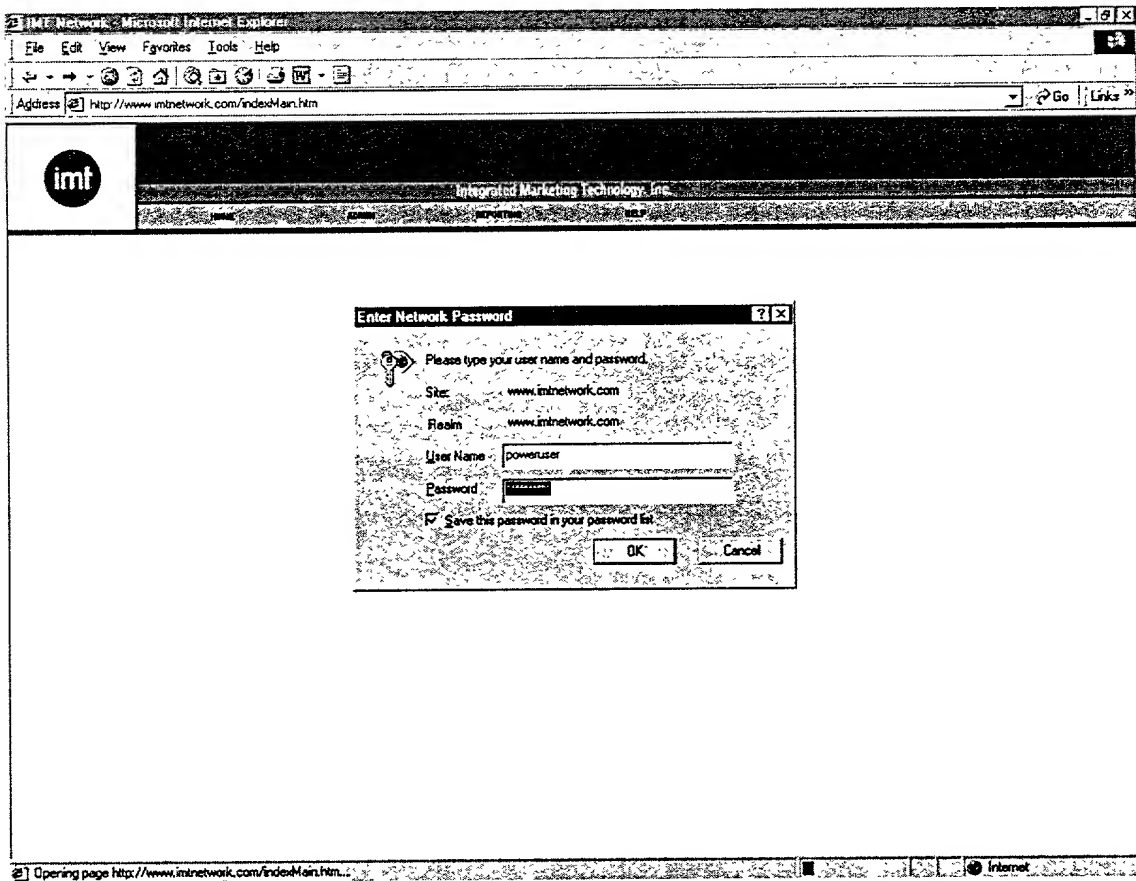


### allsetuptabs.asp

Description: This page provides a user the ability to choose a lookup table to administer. Administration is the adding, editing or deleting of records in a specific database table.



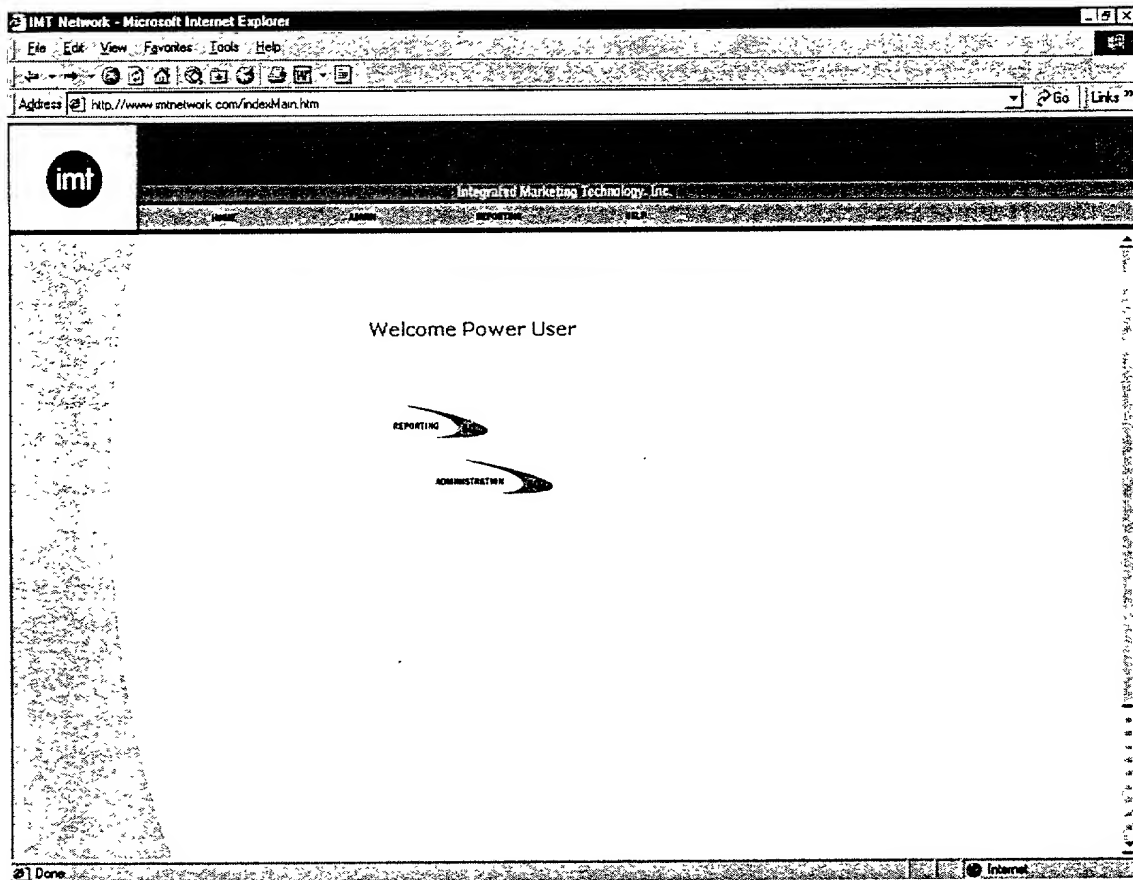
Description: This page presents a user logon dialog box requiring a valid username and password in order to gain access to the application server.





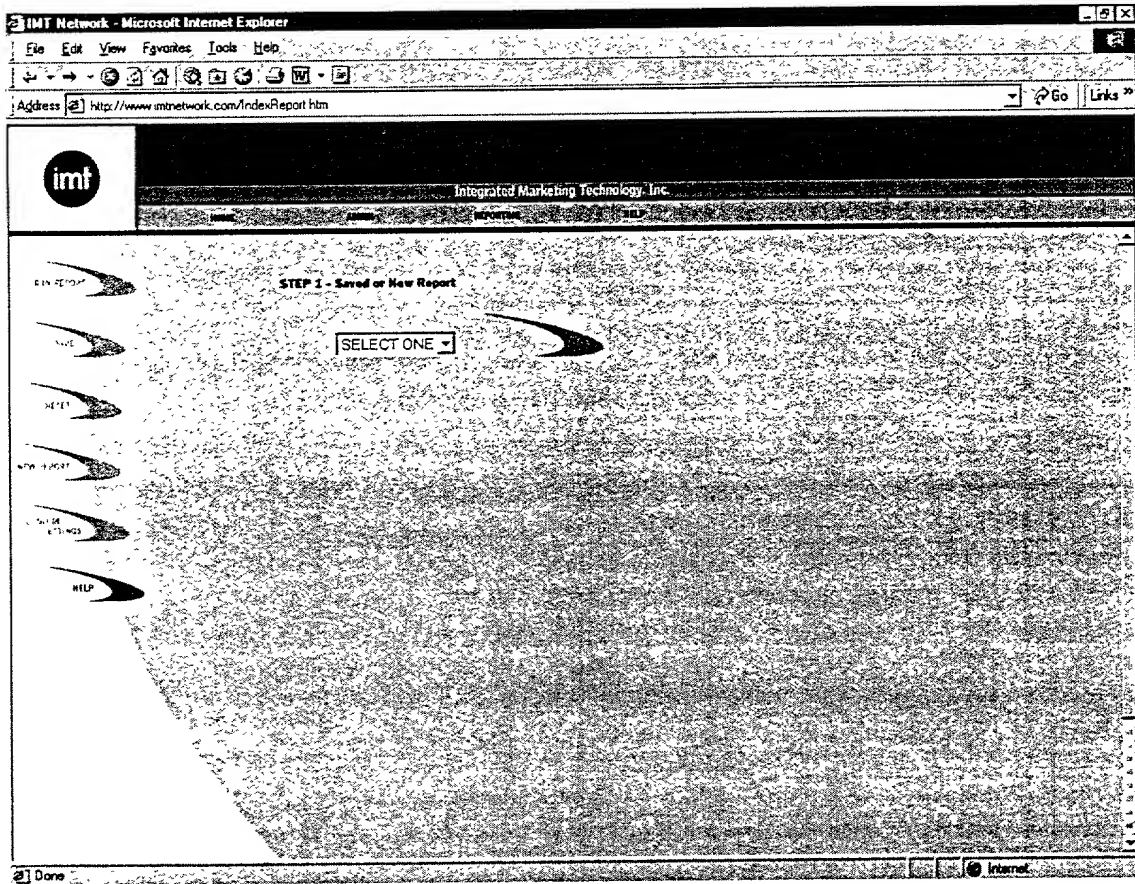
### welcome.asp

Description: This page provides a user welcome message and offers an option to choose either administrative functions or reporting functions.



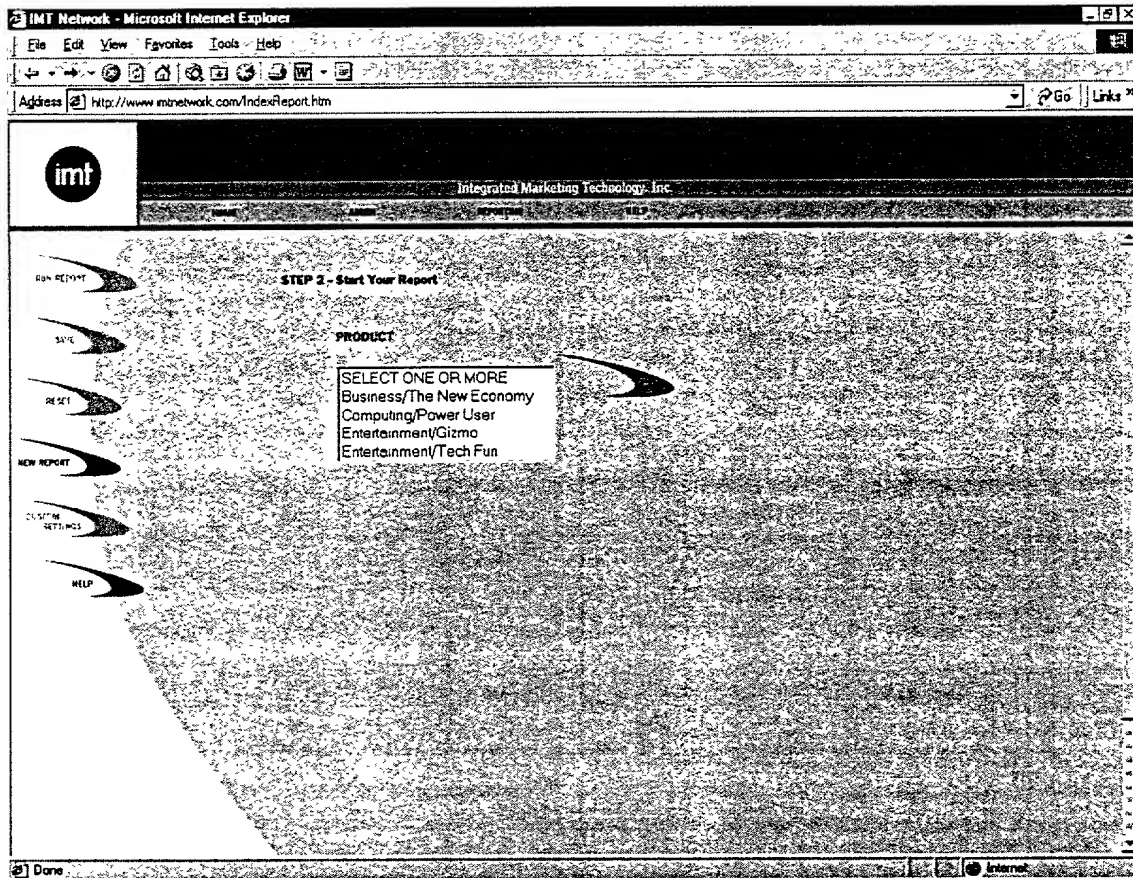
## login.asp

Description: This page provides a user the ability to choose between creating a new report or selecting a saved report by selecting a record from the `report_defs` database table.



### choosecat.asp

Description: This page provides a user the ability to choose a product(s) when creating a new report. This is the first selection in a series of selection required to create a new report.



### choosecat2.asp

Description: This page provides a user the ability to choose a report category when creating a new report. A report category is a grouping of specific reports (styles). Report categories and reports are stored in the report\_cat database table.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/IndexReport.htm Go Links

imt Integrated Marketing Technology, Inc.

STEP 2 - Start Your Report

REPORT CATEGORY

SELECT ONE

PREVIOUS SELECTIONS

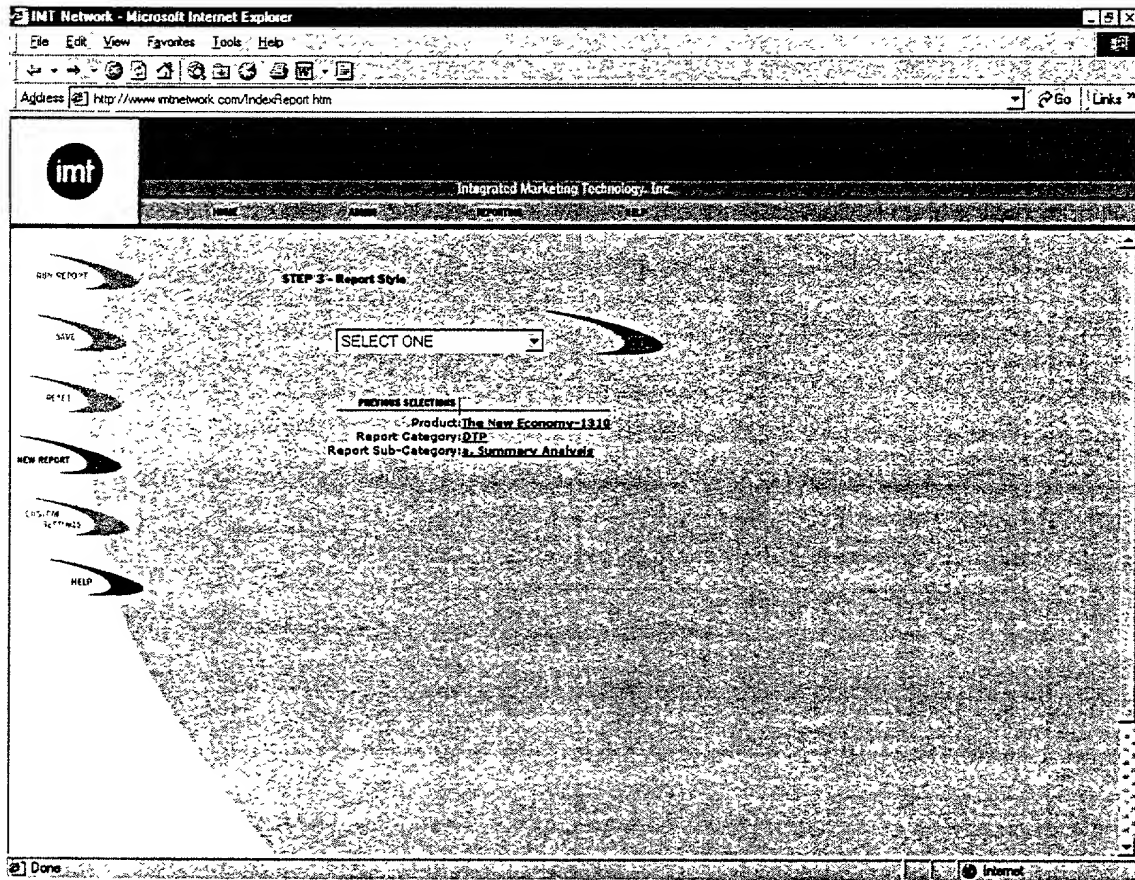
Product: The New Economy - 1216

NEW REPORT REPORT SELECT VIEW PRINT HELP

Done Internet

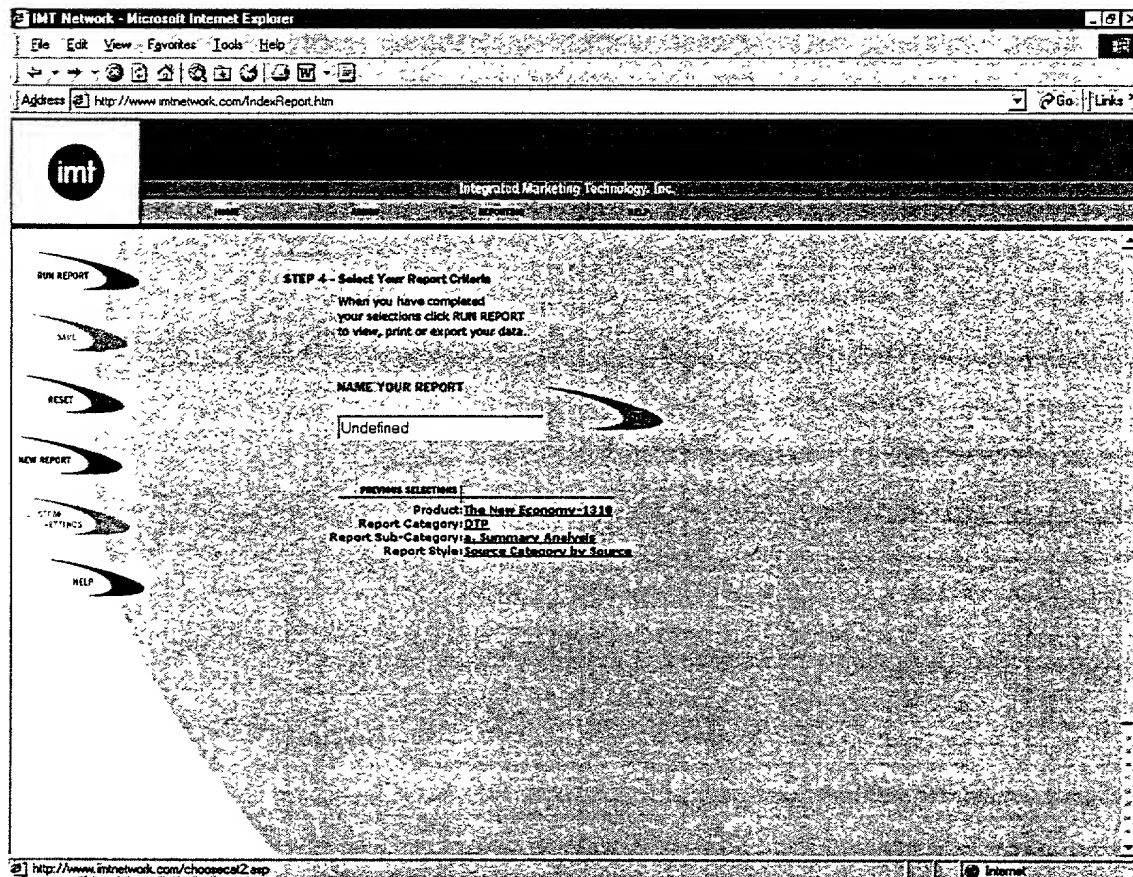
### chooserpt.asp

Description: This page provides a user the ability to choose a specific report (style) when creating a new report. A report (style) is a pre-defined view of database information.



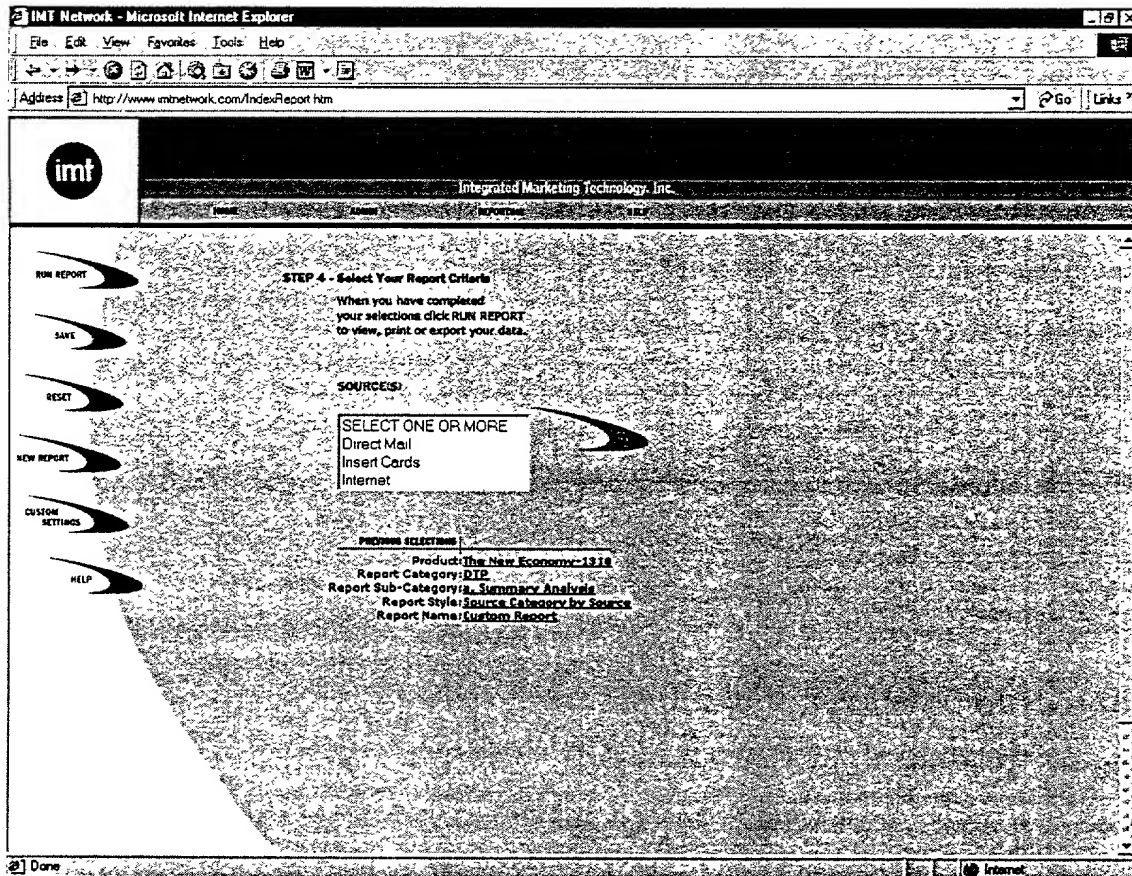
## reporta.asp

Description: This page provides a user the ability pass a text string to the report to serve as a report name. A report name appears in the header of a new report and is stored in the `report_defs` table once a report is saved.



## reportb.asp

Description: This page provides a user the ability to pass source selections to a report. A source is a grouping in IMT's proprietary database hierarchy.





## reportc.asp

Description: This page provides a user the ability to pass **campaign type** selections to a report. A campaign type is a grouping in IMT's proprietary database hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/IndexReport.htm

imt Integrated Marketing Technology, Inc.

**STEP 4 - Select Your Report Criteria**

When you have completed your selections click RUN REPORT to view, print or export your data.

CAMPAIGN TYPE(S):

All

PREVIOUS SELECTIONS

Product: The New Economy - 1210

Report Category: DTP Previous Selections

Report Sub-Category: Summary

Report Style: Source Category by Source

Report Name: Custom Report

Source(s): Direct Mail

RUN REPORT

SAVE

RESET

NEW REPORT

CUSTOM SETTINGS

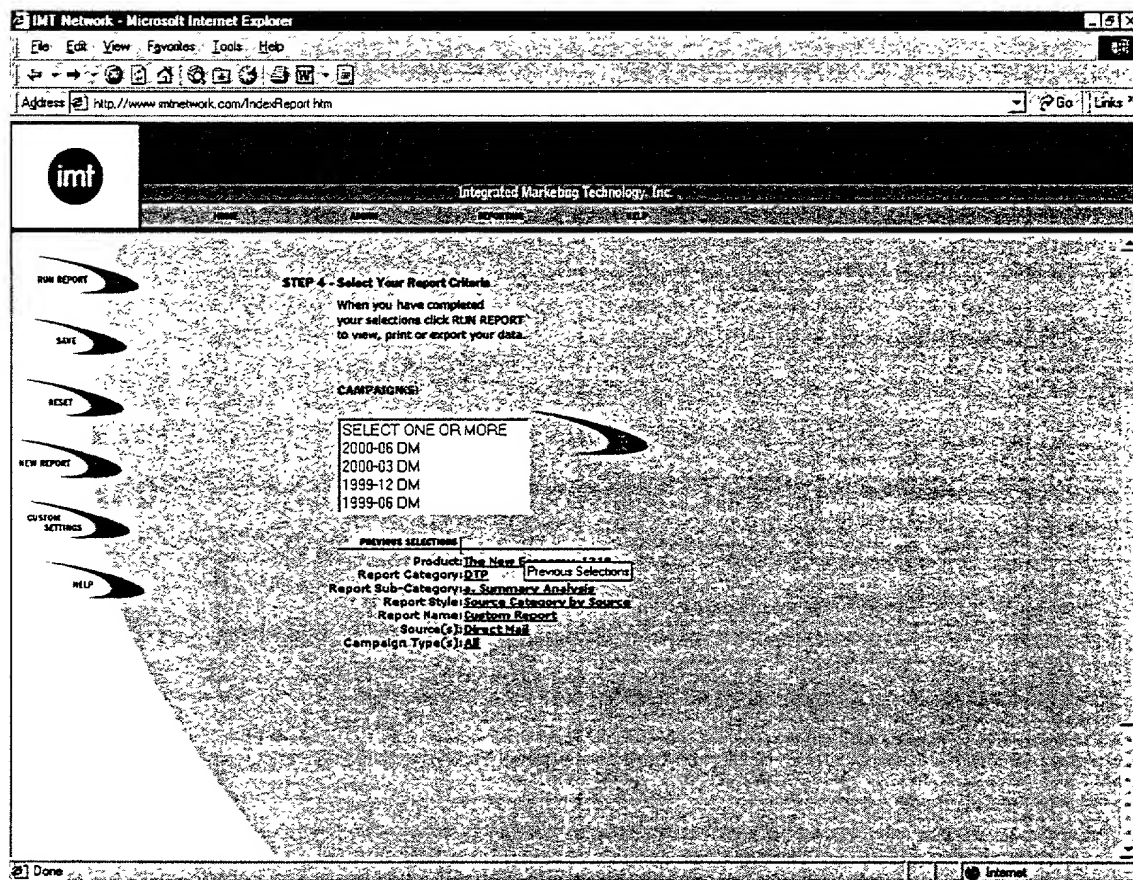
HELP

Done Internet



## reportd.asp

Description: This page provides a user the ability to pass campaign selections to a report. A campaign is a grouping in IMT's proprietary database hierarchy.



## reporte.asp

Description: This page provides a user the ability to pass panel type selections to a report. A panel type is a grouping in IMT's proprietary database hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/IndexReport.htm

Go Links

imt Integrated Marketing Technology, Inc.

RUN REPORT  
SAVE  
RESET  
NEW REPORT  
CUSTOM SETTINGS  
HELP

**STEP 4 - Select Your Report Criteria**  
When you have completed your selections click RUN REPORT to view, print or export your data.

**PANEL TYPE:**

- All
- Contrinutions
- House Files
- List Tests
- Package Test
- Retests

**PREVIOUS SELECTIONS**

Product: The New Economy: 1218  
Report Category: DTP  
Report Sub-Category: Summary Analysis  
Report Style: Source Category by Source  
Report Name: Custom Report  
Source(s): Direct Mail  
Campaign Type(s): AD  
2888-88 DM, 2888-88 DM, 1818  
Campaign(s): 12 DM

http://www.imtnetwork.com/choosereport.asp

## reportf.asp

Description: This page provides a user the ability to pass panel test type selections to a report. A panel test type is a grouping in IMT's proprietary database hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/IndexReport.htm

imt Integrated Marketing Technology, Inc.

Home Admin Reporting Help

RUN REPORT  
SAVE  
RESET  
NEW REPORT  
CUSTOM SETTINGS  
HELP

**STEP 4 - Select Your Report Criteria**  
When you have completed your selections click RUN REPORT to view, print or export your data.

PANEL TEST TYPES

- All
- Control
- Control Test
- Copy Test
- List Test
- Offer Test
- Package Test
- Premium Test
- Price Test
- Timing Test

PREVIOUS SELECTIONS

- Product: The New Economy-1318
- Report Category: DTP
- Report Sub-Category: Summary Analysis
- Report Style: Source Category by Source
- Report Name: System Report
- Source(s): Direct Mail
- Campaign Type(s): All
- Campaign(s): 2000-06 DM, 2000-03 DM, 1999-12 DM
- Panel Types: All

Done Internet

## reportg.asp

Description: This page provides a user the ability to pass panel selections to a report. A panel is a grouping in IMT's proprietary database hierarchy.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/IndexReport.htm Go Links

imt Integrated Marketing Technology, Inc.

STEP 4 - Select Your Report Criteria

When you have completed your selections click RUN REPORT to view, print or export your data.

PANEL NAME(S):

All

01 Continuations - 12/\$12

01 Continuations - 26/\$12

02 Retests - 12/\$12

02 Retests - 26/\$12

03 List Refinements - 26/\$12

03 List Tests - 12/\$12

03 List Tests - 26/\$12

04 House Files - 12/\$12

04 House Files - 26/\$12

PREVIOUS SELECTIONS

Product: The New Economy - 1318

Report Category: DTP

Report Sub-Category: Summary Analysis

Report Style: Source Category by Source

Report Name: Custom Report

Source(s): Direct Mail

Campaign Type(s): All

Campaign(s): 2000-05 DM, 2000-03 DM, 1999-12 DM

Panel Types: All

Panel Test Types: All

Done Internet

### saverpt.asp

Description: This page provides a user the ability to save a set of report selections by adding a record to the report\_defs table. The report\_defs table stores selections about division, group, product, report group, report, report name, report location and report selections.

IMT Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.imtnetwork.com/IndexReport.htm

imt Integrated Marketing Technology, Inc.

RUN REPORT

NEW REPORT

SYSTEM SETTINGS

HELP

TO SAVE: <Custom Report>

SELECT A REPORT GROUP

SELECT ONE

CREATE A NEW REPORT GROUP

Enter new group name here!

PREVIOUS SELECTIONS

Division/Group: Magazine/Business

Product: The New Economy

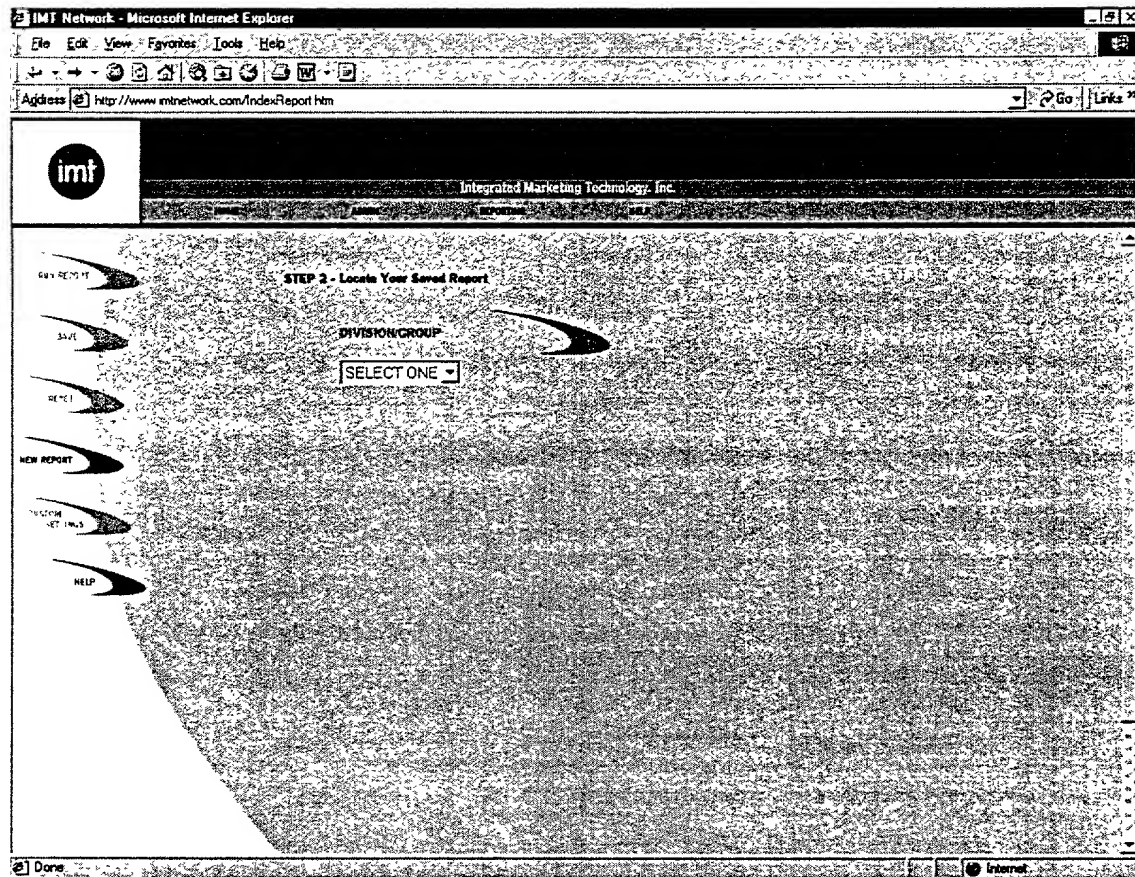
Report Category: DTP

Done



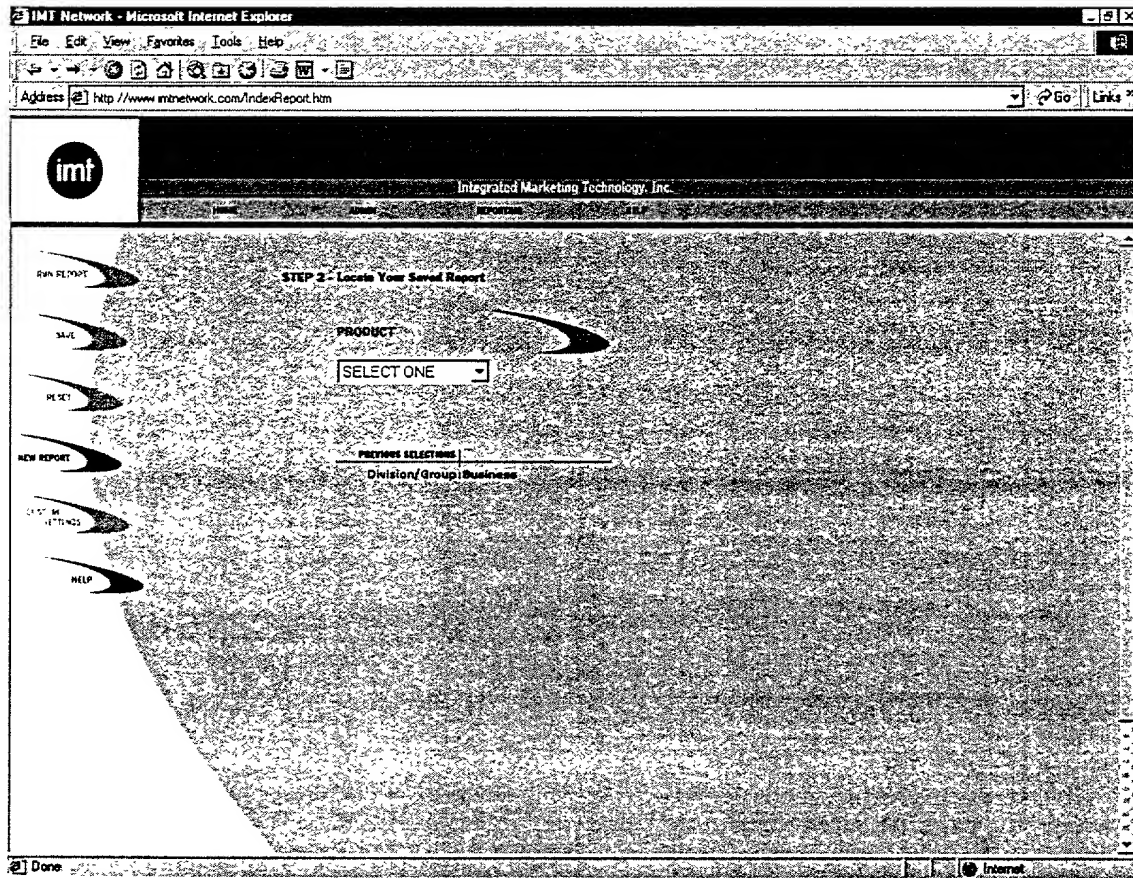
### choosesaved.asp

Description: This page provides a user the ability to locate a saved report from the **report\_defs** table by filtering selections by **division** and **group**. Both **division** and **group** are groupings in IMT's proprietary database hierarchy



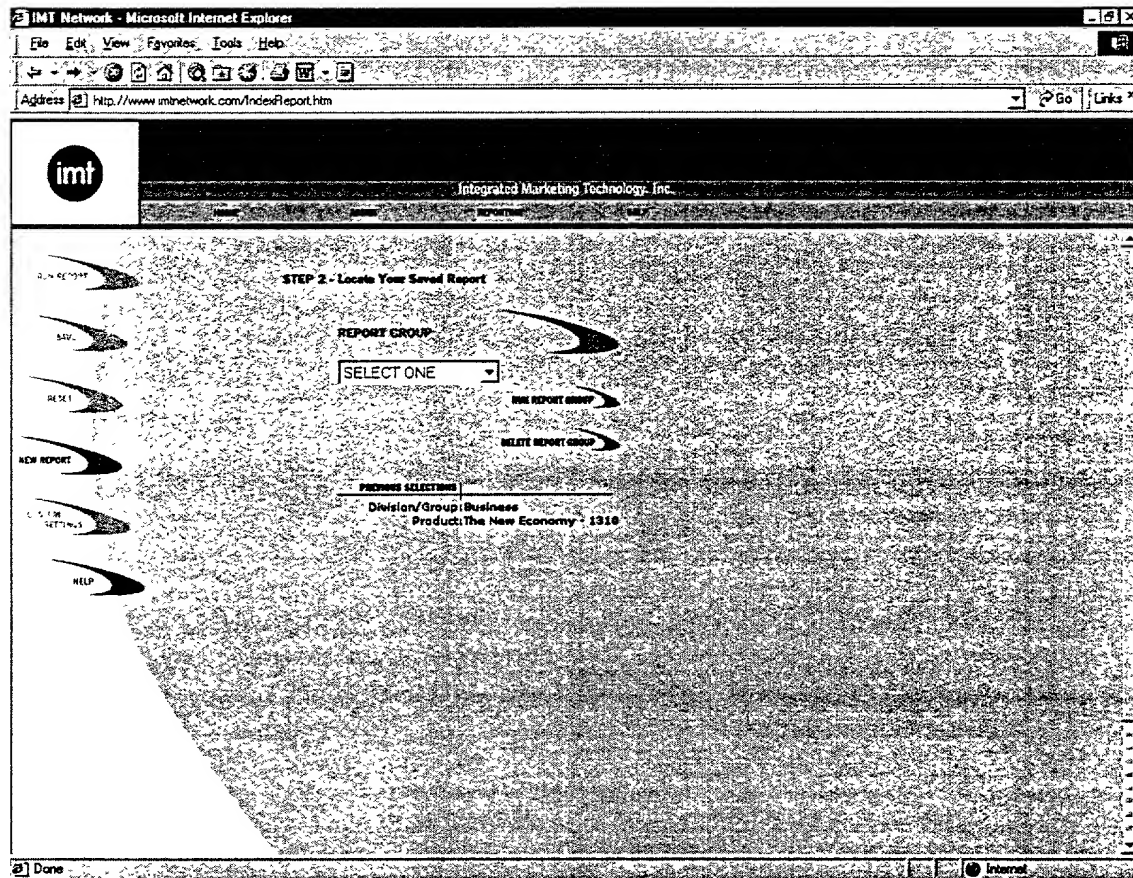
### choosesavedb.asp

Description: This page provides a user the ability to locate a saved report from the **report\_defs** table by filtering selections by **product**. A **product** is a grouping in IMT's proprietary database hierarchy



### choosesavedc.asp

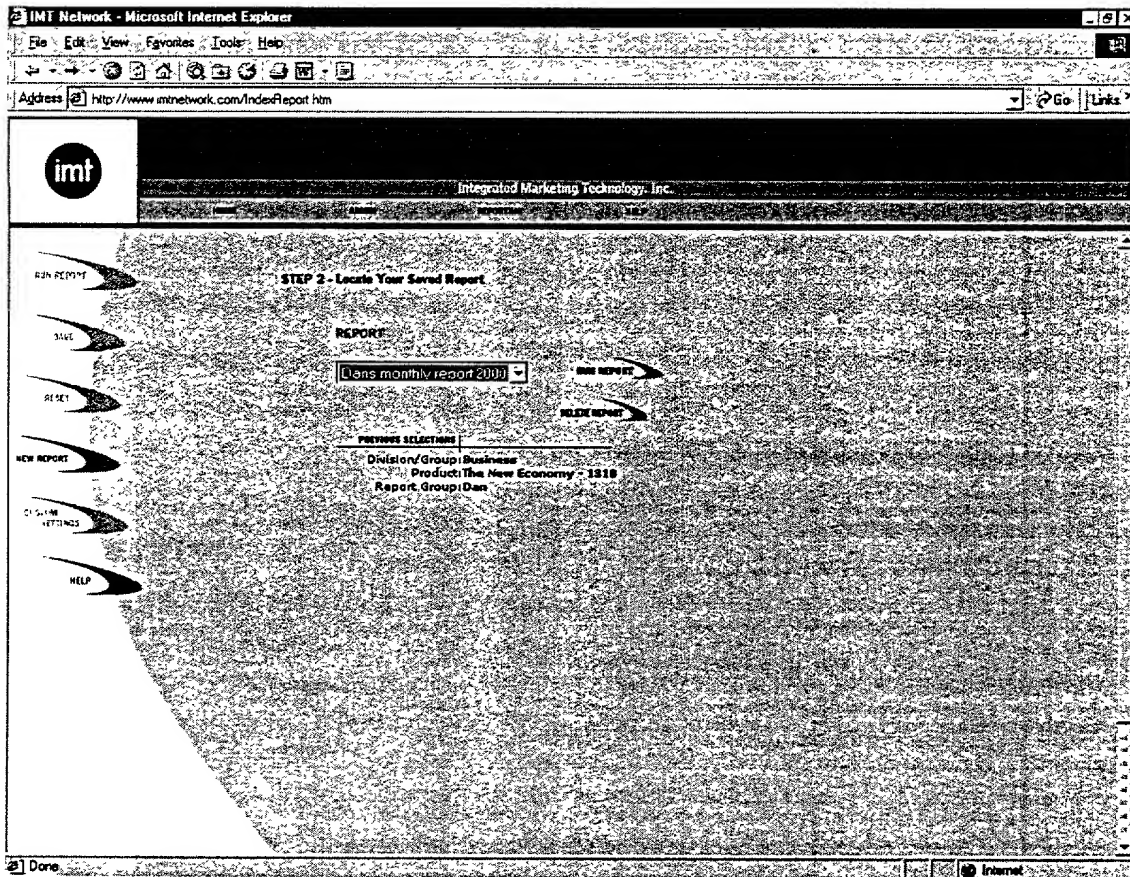
Description: This page provides a user the ability to locate a saved report from the **report\_defs** table by filtering selections by **report group**. A **report group** is a grouping of saved reports.





### choosesavedd.asp

Description: This page provides a user the ability to locate and execute a saved report from the report\_defs table by filtering selections by report name. A report name is the unique identifier of a record in the report\_defs table.





## DATABASE TABLES (in alpha order)

- agent\_defs
- agents\_summary
- autoseup\_campaign\_def
- autoseup\_panel\_def
- autoseup\_renewal\_psdef
- autoseup\_scheme
- avail\_keys
- campaigns
- corp\_info
- corp\_panel\_sub\_type
- corp\_panel\_test\_type
- corp\_panel\_type
- effort\_sequence
- est\_max\_effort
- expense\_sets
- keys
- list\_cats
- list\_names
- list\_segments
- panels
- prior\_source\_desc
- prod\_camp\_status
- prod\_camp\_type
- prod\_panel\_package
- r101
- r101\_flow
- r101\_toplevel
- r401\_flow
- r401\_summary
- rankings\_base\_info
- revenue\_set
- rpt\_cats
- rpt\_defs
- source
- source\_descriptions
- src\_category
- src\_channel
- starts
- summary\_log



Table Name	Field Name	Type	Length	Precision	Nullable
agent_defs	rec_id	numeric	9	18	N
agent_defs	fulfillment_house	varchar	250	250	N
agent_defs	agent_code	varchar	50	50	N
agent_defs	agent_description	varchar	250	250	Y
agent_defs	agent_name	varchar	30	30	Y
agents_summary	agent_rec_num	numeric	9	18	N
agents_summary	product_id	decimal	9	18	N
agents_summary	agent_source	varchar	50	50	Y
agents_summary	bus_month_date	smalldatetime	4	16	N
agents_summary	agent_reorg_date	smalldatetime	4	16	N
agents_summary	agent_key	varchar	12	12	Y
agents_summary	agent_code	varchar	50	50	N
agents_summary	fulfillment_house	varchar	250	250	Y
agents_summary	agent_total_subs	numeric	9	18	Y
agents_summary	agent_renew_subs	numeric	9	18	Y
agents_summary	agent_copies	numeric	9	18	Y
agents_summary	agent_gross_revenue	money	8	19	Y
agents_summary	agent_net_revenue	money	8	19	Y
agents_summary	agent_remit	numeric	9	14	Y
agents_summary	agent_cash_cncl_subs	numeric	9	18	Y
agents_summary	agent_cash_cncl_grs_dllr	money	8	19	Y
agents_summary	agent_cash_cncl_net_dllr	money	8	19	Y
agents_summary	agent_cash_cncl_issues	numeric	9	18	Y
agents_summary	agent_reinstate_subs	numeric	9	18	Y
agents_summary	agent_reinstate_grs_dllr	money	8	19	Y
agents_summary	agent_reinstate_net_dllr	money	8	19	Y
agents_summary	agent_reinstate_issues	numeric	9	18	Y
agents_summary	agent_avg_tot_net_term	numeric	9	18	Y
agents_summary	agent_avg_tot_net_revenue	money	8	19	Y
agents_summary	import_file_type	char	1	1	Y
autosetup_campaign_def	product_id	varchar	50	50	N
autosetup_campaign_def	campaign_code	varchar	50	50	N
autosetup_campaign_def	campaign_description	varchar	200	200	N
autosetup_campaign_def	src_cat_name	varchar	50	50	N
autosetup_campaign_def	campaign_date	datetime	8	23	Y
autosetup_campaign_def	panel_scheme	varchar	50	50	Y
autosetup_campaign_def	rec_id	int	4	10	N
autosetup_panel_def	product_id	int	4	10	N
autosetup_panel_def	panel_scheme	varchar	50	50	N
autosetup_panel_def	panel_codes	varchar	50	50	Y
autosetup_panel_def	panel_name	varchar	50	50	Y
autosetup_panel_def	rec_id	int	4	10	N
autosetup_renewal_psdef	product_id	varchar	50	50	N
autosetup_renewal_psdef	title	varchar	50	50	N
autosetup_renewal_psdef	src_cat_name	varchar	50	50	N
autosetup_renewal_psdef	source_name	varchar	50	50	N



Table Name	Field Name	Type	Length	Precision	Nullable
autosetup_renewal_psdef	source_indicator	varchar	50	50	N
autosetup_renewal_psdef	prior_source_indicator	varchar	50	50	N
autosetup_renewal_psdef	valid_from_date	datetime	8	23	N
autosetup_renewal_psdef	valid_to_date	datetime	8	23	N
autosetup_renewal_psdef	rec_id	int	4	10	N
autosetup_scheme	product_id	varchar	50	50	N
autosetup_scheme	title	varchar	50	50	N
autosetup_scheme	grp_name	varchar	50	50	N
autosetup_scheme	src_cat_name	varchar	50	50	N
autosetup_scheme	src_indicator	varchar	50	50	N
autosetup_scheme	source_name	varchar	50	50	N
autosetup_scheme	valid_from_date	datetime	8	23	N
autosetup_scheme	valid_to_date	datetime	8	23	N
autosetup_scheme	setup_mask	varchar	50	50	Y
autosetup_scheme	e_set_name	varchar	50	50	Y
autosetup_scheme	rec_id	int	4	10	N
avail_keys	product_id	numeric	5	8	N
avail_keys	full_promo_key	varchar	25	25	N
avail_keys	mail_qty	numeric	9	12	Y
avail_keys	dt_valid_to	smalldatetime	4	16	Y
avail_keys	rsrvrd	varchar	25	25	Y
avail_keys	source_identifier	varchar	5	5	Y
avail_keys	position1	varchar	3	3	Y
avail_keys	position2	char	1	1	Y
avail_keys	position3	char	1	1	Y
avail_keys	position4	char	1	1	Y
avail_keys	position5	char	1	1	Y
avail_keys	position6	char	1	1	Y
avail_keys	position7	char	1	1	Y
avail_keys	position8	char	1	1	Y
avail_keys	position9	char	1	1	Y
avail_keys	position10	char	1	1	Y
avail_keys	rec_id	numeric	9	18	N
avail_keys	fulfillment_house	varchar	50	50	Y
campaigns	src_indicator	char	3	3	N
campaigns	src_cat_name	varchar	30	30	N
campaigns	product_id	numeric	5	8	N
campaigns	camp_type	varchar	50	50	Y
campaigns	camp_name	varchar	250	250	N
campaigns	camp_status_name	varchar	20	20	Y
campaigns	agt_code	varchar	10	10	Y
campaigns	agt_descr	varchar	50	50	Y
campaigns	camp_date_fr	datetime	8	23	Y
campaigns	camp_date_to	datetime	8	23	Y
campaigns	e_set_name	varchar	50	50	Y
campaigns	r_set_name	varchar	50	50	Y
campaigns	grp_name	varchar	50	50	Y

Table Name	Field Name	Type	Length	Precision	Nullable
campaigns	title_name	varchar	50	50	Y
campaigns	source_name	varchar	250	250	N
campaigns	rec_id	numeric	9	18	N
campaigns	Newstand_Sale	numeric	9	18	Y
campaigns	Budgeted_vol	numeric	9	18	Y
campaigns	Budgeted_gross_pct	numeric	9	18	Y
campaigns	campaign_code	varchar	50	50	Y
corp_info	product_id	numeric	9	18	N
corp_info	div_name	varchar	50	50	N
corp_info	grp_name	varchar	50	50	N
corp_info	title_name	varchar	50	50	N
corp_info	corp_info_descr	varchar	250	250	Y
corp_info	rec_id	numeric	9	18	N
corp_panel_sub_type	rec_id	numeric	9	18	N
corp_panel_sub_type	panel_sub_type_name	varchar	50	50	N
corp_panel_sub_type	panel_sub_type_descr	varchar	250	250	Y
corp_panel_test_type	rec_id	numeric	9	18	N
corp_panel_test_type	panel_test_type_name	varchar	50	50	N
corp_panel_test_type	panel_test_type_descr	varchar	250	250	Y
corp_panel_type	rec_id	numeric	9	18	N
corp_panel_type	panel_type_name	varchar	50	50	N
corp_panel_type	panel_type_descr	varchar	250	250	Y
dtproperties	id	int	4	10	N
dtproperties	objectid	int	4	10	Y
dtproperties	property	varchar	64	64	N
dtproperties	value	varchar	255	255	Y
dtproperties	lvalue	image	16		Y
dtproperties	version	int	4	10	N
EFFORT_SEQUENCE	Product_id	numeric	9	18	Y
EFFORT_SEQUENCE	Effort_indicator	varchar	1	1	Y
EFFORT_SEQUENCE	effort_sequence	numeric	9	18	Y
EFFORT_SEQUENCE	effort_desc	varchar	50	50	Y
Est_max_effort	Product_id	decimal	9	18	Y
Est_max_effort	source_identifier	varchar	50	50	Y
Est_max_effort	issue_identifier	varchar	50	50	Y
Est_max_effort	prior_source_identifier	varchar	50	50	Y
Est_max_effort	max_effort	char	1	1	Y
Est_max_effort	min_sum_mail_qty	numeric	9	18	Y
expense_sets	product_id	numeric	9	18	N
expense_sets	e_set_name	varchar	50	50	N
expense_sets	print_cost	numeric	9	11	Y
expense_sets	print_cpm_cpu	varchar	10	10	Y
expense_sets	print_basis	varchar	20	20	Y
expense_sets	merge_purge	numeric	9	11	Y
expense_sets	merge_cpm_cpu	varchar	10	10	Y
expense_sets	merge_basis	varchar	20	20	Y
expense_sets	ltr_cost	numeric	9	11	Y



Table Name	Field Name	Type	Length	Precision	Nullable
expense_sets	ltr_cpm_cpu	varchar	10	10	Y
expense_sets	ltr_basis	varchar	20	20	Y
expense_sets	postage_out_cost	numeric	9	11	Y
expense_sets	postage_out_cpm_cpu	varchar	10	10	Y
expense_sets	postage_out_basis	varchar	20	20	Y
expense_sets	postage_in_cost	numeric	9	11	Y
expense_sets	postage_in_cpm_cpu	varchar	10	10	Y
expense_sets	postage_in_basis	varchar	20	20	Y
expense_sets	premium_cost	numeric	9	11	Y
expense_sets	premium_cpm_cpu	varchar	10	10	Y
expense_sets	premium_basis	varchar	20	20	Y
expense_sets	badpay_cost	numeric	9	11	Y
expense_sets	badpay_cpm_cpu	varchar	10	10	Y
expense_sets	badpay_basis	varchar	20	20	Y
expense_sets	billing_cost	numeric	9	11	Y
expense_sets	billing_cpm_cpu	varchar	10	10	Y
expense_sets	billing_basis	varchar	20	20	Y
expense_sets	subs_svc_costs	numeric	9	11	Y
expense_sets	subs_svc_cpm_cpu	varchar	10	10	Y
expense_sets	subs_svc_basis	varchar	20	20	Y
expense_sets	exp_fixed_costs	numeric	9	11	Y
expense_sets	exp_fixed_cpm_cpu	varchar	10	10	Y
expense_sets	exp_fixed_basis	varchar	20	20	Y
expense_sets	exp_other_costs	numeric	9	11	Y
expense_sets	exp_other_cpm_cpu	varchar	10	10	Y
expense_sets	exp_other_basis	varchar	20	20	Y
expense_sets	rec_id	numeric	9	18	N
keys	full_promo_key	varchar	25	25	N
keys	dt_valid_from	smalldatetime	4	16	N
keys	panel_name	varchar	250	250	N
keys	camp_name	varchar	250	250	N
keys	dt_valid_to	smalldatetime	4	16	N
keys	src_indicator	char	3	3	N
keys	src_cat_name	varchar	30	30	N
keys	product_id	numeric	9	18	N
keys	e_set_name	varchar	50	50	Y
keys	list_id	numeric	9	10	Y
keys	level1_expense	numeric	9	12	Y
keys	list_cost_rollout_cpm	numeric	9	14	Y
keys	list_cost_actual_cpm	numeric	9	14	Y
keys	fixed_expense	numeric	9	18	Y
keys	subtotal_flag	varchar	6	6	N
keys	level2_expense	numeric	9	12	Y
keys	r_set_name	varchar	50	50	Y
keys	level3_expense	numeric	9	12	Y
keys	level1_revenue	numeric	9	12	Y
keys	level2_revenue	numeric	9	12	Y



Table Name	Field Name	Type	Length	Precision	Nullable
keys	level3_revenue	numeric	9	12	Y
keys	p_key_descr	varchar	250	250	Y
keys	panel_type	varchar	50	50	Y
keys	panel_subtype	varchar	50	50	Y
keys	test_type	varchar	50	50	Y
keys	panel_offer	varchar	20	20	Y
keys	panel_offer_price	money	8	19	Y
keys	panel_pkg	varchar	50	50	Y
keys	panel_premium	varchar	20	20	Y
keys	panel_date_start	datetime	8	23	Y
keys	panel_date_end	datetime	8	23	Y
keys	grp_name	varchar	50	50	Y
keys	title_name	varchar	50	50	Y
keys	source_name	varchar	250	250	Y
keys	camp_type	varchar	50	50	Y
keys	camp_status_name	varchar	20	20	Y
keys	agt_code	varchar	10	10	Y
keys	agt_descr	varchar	50	50	Y
keys	camp_date_fr	datetime	8	23	Y
keys	camp_date_to	datetime	8	23	Y
keys	key_mail_qty	numeric	9	18	Y
keys	mail_qty_override	decimal	9	18	Y
keys	merge_purge_qty	numeric	9	18	Y
keys	list_cost_basis	varchar	25	25	Y
keys	list_cost_factor	decimal	9	18	Y
keys	list_cat_name	varchar	250	250	Y
keys	list_name	varchar	250	250	Y
keys	list_segment_name	varchar	250	250	Y
keys	rec_id	numeric	9	18	N
keys	optional_effort_id	varchar	50	50	Y
keys	Newstand_Sale	numeric	9	18	Y
keys	newstand_sale_marker	char	1	1	Y
keys	Budgeted_gross_pct	numeric	9	18	Y
keys	Budgeted_vol	numeric	9	18	Y
keys	date_assigned	datetime	8	23	Y
keys	campaign_code	varchar	50	50	Y
keys	panel_code	varchar	50	50	Y
keys	dbusername	varchar	50	50	Y
keys	order_qty	int	4	10	Y
keys	dist_qty	int	4	10	Y
list_cats	product_id	numeric	9	18	N
list_cats	list_cat_name	varchar	250	250	N
list_cats	list_cat_descr	varchar	250	250	Y
list_cats	rec_id	numeric	9	18	N
list_names	product_id	numeric	9	18	N
list_names	list_name	varchar	250	250	N
list_names	list_name_description	varchar	250	250	Y



## IMT Subscription Marketing Reporting System

Table Name	Field Name	Type	Length	Precision	Nullable
list_names	rec_id	numeric	9	18	N
list_segments	product_id	numeric	9	18	N
list_segments	list_segment_name	varchar	250	250	N
list_segments	list_segment_description	varchar	250	250	Y
list_segments	rec_id	numeric	9	18	N
logs	product_id	varchar	250	250	N
logs	login	varchar	50	50	N
logs	log_timestamp	datetime	8	23	N
logs	action	varchar	50	50	Y
logs	action_description	varchar	255	255	Y
logs	failed_flag	varchar	10	10	Y
logs	trace_on	varchar	10	10	Y
mods	ticket_id	numeric	9	18	N
mods	mod_app_name	varchar	50	50	N
mods	mod_date	smalldatetime	4	16	Y
mods	mod_description	varchar	250	250	N
mods	mod_notes	varchar	250	250	Y
panels	camp_name	varchar	250	250	N
panels	src_indicator	char	3	3	N
panels	panel_name	varchar	250	250	N
panels	src_cat_name	varchar	30	30	N
panels	product_id	numeric	5	8	N
panels	panel_type	varchar	50	50	Y
panels	e_set_name	varchar	50	50	Y
panels	panel_subtype	varchar	50	50	Y
panels	r_set_name	varchar	50	50	Y
panels	test_type	varchar	50	50	Y
panels	panel_offer	varchar	20	20	Y
panels	panel_offer_price	money	8	19	Y
panels	panel_pkg	varchar	50	50	Y
panels	panel_premium	varchar	20	20	Y
panels	panel_date_start	datetime	8	23	Y
panels	panel_date_end	datetime	8	23	Y
panels	grp_name	varchar	50	50	Y
panels	title_name	varchar	50	50	Y
panels	source_name	varchar	250	250	N
panels	camp_type	varchar	50	50	Y
panels	camp_status_name	varchar	20	20	Y
panels	agt_code	varchar	10	10	Y
panels	agt_descr	varchar	50	50	Y
panels	camp_date_fr	datetime	8	23	Y
panels	camp_date_to	datetime	8	23	Y
panels	rec_id	numeric	9	18	N
panels	newstand_sale_marker	char	1	1	Y
panels	Newstand_Sale	numeric	9	18	Y
panels	Budgeted_gross_pct	numeric	9	18	Y
panels	Budgeted_vol	numeric	9	18	Y





# IMT Subscription Marketing Reporting System

Table Name	Field Name	Type	Length	Precision	Nullable
panels	campaign_code	varchar	50	50	Y
panels	panel_code	varchar	50	50	Y
panels	order_qty	int	4	10	Y
panels	dist_qty	int	4	10	Y
Prior_source_desc	product_id	numeric	5	9	N
Prior_source_desc	prior_source_identifier	varchar	2	2	N
Prior_source_desc	prior_source_desc	varchar	20	20	Y
Prior_source_desc	valid_from_date	datetime	8	23	Y
Prior_source_desc	valid_to_date	datetime	8	23	Y
Prior_source_desc	rec_id	numeric	9	18	N
Prior_source_desc	billing_cur_src_desc	varchar	255	255	Y
Prior_source_desc	agent_src_desc	varchar	50	50	Y
Prior_source_desc	model_source	varchar	50	50	Y
Prior_source_desc	model_source_cat	varchar	50	50	Y
prod_camp_status	rec_id	numeric	9	18	N
prod_camp_status	product_id	numeric	9	18	N
prod_camp_status	camp_status_name	varchar	50	50	N
prod_camp_status	camp_status_descr	varchar	250	250	Y
prod_camp_type	rec_id	numeric	9	18	N
prod_camp_type	product_id	numeric	9	18	N
prod_camp_type	camp_type_name	varchar	50	50	N
prod_camp_type	camp_type_descr	varchar	250	250	Y
prod_panel_package	product_id	numeric	9	18	N
prod_panel_package	package_name	varchar	50	50	N
prod_panel_package	package_description	varchar	250	250	Y
prod_panel_package	rec_id	numeric	9	18	N
r101	rct_rec_num	decimal	9	18	N
r101	r101_reorg_date	smalldatetime	4	16	N
r101	product_id	numeric	9	18	N
r101	r101_key	varchar	25	25	N
r101	source_identifier	varchar	50	50	Y
r101	issue_identifier	varchar	3	3	Y
r101	prior_source_identifier	char	2	2	Y
r101	effort_identifier	char	1	1	Y
r101	split_identifier	varchar	10	10	Y
r101	mail_qty	numeric	9	18	Y
r101	select_date	smalldatetime	4	16	Y
r101	key_descr	varchar	50	50	Y
r101	subs_this_week	numeric	9	18	Y
r101	subs_to_date	numeric	9	18	Y
r101	gross_subs	numeric	9	18	Y
r101	gross_subs_pct	decimal	5	6	Y
r101	net_subs	numeric	9	18	Y
r101	net_subs_pct	decimal	5	6	Y
r101	tot_net_subs	numeric	9	18	Y
r101	tot_net_subs_pct	decimal	5	6	Y
r101	pct_credit	numeric	9	18	Y



Table Name	Field Name	Type	Length	Precision	Nullable
r101	pct_cred_renew	decimal	5	6	Y
r101	cred_pay_amt	decimal	5	6	Y
r101	cred_pay_pct	numeric	9	18	Y
r101	avg_net_trm	decimal	5	6	Y
r101	avg_net_trm_value	decimal	5	6	Y
r101	avg_tot_net_trm	numeric	9	18	Y
r101	avg_tot_net_trm_value	numeric	9	18	Y
r101	subs_phone	numeric	9	18	Y
r101	subs_c_card	numeric	9	18	Y
r101	install_1st	int	4	10	Y
r101	install_1st_total	int	4	10	Y
r101	subs_gift	int	4	10	Y
r101	subs_gift_pct_1st_trm	decimal	5	6	Y
r101	sub_years_gross	decimal	9	12	Y
r101	sub_years_gross_tot_net	decimal	9	12	Y
r101	gross_sub_revenue	decimal	9	18	Y
r101	net_sub_revenue	decimal	9	18	Y
r101	gross_sub_copies	decimal	9	18	Y
r101	net_sub_copies	decimal	9	18	Y
r101	cash_subs	numeric	9	18	Y
r101	est_max_eff_mail_qty	decimal	9	12	Y
r101	gross_cred_subs	numeric	9	18	Y
r101	net_cred_subs	decimal	9	18	Y
r101	fulfillment_title_name	varchar	50	50	Y
r101	gross_list_rental_qty	decimal	9	10	Y
r101	net_list_rental_qty	decimal	9	10	Y
r101	package_identifier	char	1	1	Y
r101	key_descr2	varchar	50	50	Y
r101	donees_todate	numeric	9	18	Y
r101	total_orders	numeric	9	18	Y
r101	tot_order_pct_rspnce	numeric	9	18	Y
r101	renew_subs_pct	numeric	9	18	Y
r101	fulfillment_house	varchar	50	50	Y
r101_flow	rct_rec_num	decimal	9	18	N
r101_flow	r101_reorg_date	smalldatetime	4	16	N
r101_flow	product_id	numeric	9	18	N
r101_flow	r101_key	varchar	25	25	N
r101_flow	source_identifier	varchar	50	50	Y
r101_flow	issue_identifier	varchar	3	3	Y
r101_flow	prior_source_identifier	char	2	2	Y
r101_flow	effort_identifier	char	1	1	Y
r101_flow	split_identifier	varchar	10	10	Y
r101_flow	mail_qty	numeric	9	18	Y
r101_flow	select_date	smalldatetime	4	16	Y
r101_flow	key_descr	varchar	50	50	Y
r101_flow	subs_this_week	numeric	9	18	Y
r101_flow	subs_to_date	numeric	9	18	Y

Table Name	Field Name	Type	Length	Precision	Nullable
r101_flow	gross_subs	numeric	9	18	Y
r101_flow	gross_subs_pct	decimal	5	6	Y
r101_flow	net_subs	numeric	9	18	Y
r101_flow	net_subs_pct	decimal	5	6	Y
r101_flow	tot_net_subs	numeric	9	18	Y
r101_flow	tot_net_subs_pct	decimal	5	6	Y
r101_flow	pct_credit	numeric	9	18	Y
r101_flow	pct_cred_renew	decimal	5	6	Y
r101_flow	cred_pay_amt	decimal	5	6	Y
r101_flow	cred_pay_pct	numeric	9	18	Y
r101_flow	avg_net_trm	decimal	5	6	Y
r101_flow	avg_net_trm_value	decimal	5	6	Y
r101_flow	avg_tot_net_trm	numeric	9	18	Y
r101_flow	avg_tot_net_trm_value	numeric	9	18	Y
r101_flow	subs_phone	numeric	9	18	Y
r101_flow	subs_c_card	numeric	9	18	Y
r101_flow	install_1st	int	4	10	Y
r101_flow	install_1st_total	int	4	10	Y
r101_flow	subs_gift	int	4	10	Y
r101_flow	subs_gift_pct_1st_trm	decimal	5	6	Y
r101_flow	sub_years_gross	decimal	9	12	Y
r101_flow	sub_years_gross_tot_net	decimal	9	12	Y
r101_flow	gross_sub_revenue	decimal	9	18	Y
r101_flow	net_sub_revenue	decimal	9	18	Y
r101_flow	gross_sub_copies	decimal	9	18	Y
r101_flow	net_sub_copies	decimal	9	18	Y
r101_flow	cash_subs	numeric	9	18	Y
r101_flow	est_max_eff_mail_qty	decimal	9	12	Y
r101_flow	gross_cred_subs	numeric	9	18	Y
r101_flow	net_cred_subs	decimal	9	18	Y
r101_flow	fulfillment_title_name	varchar	50	50	Y
r101_flow	gross_list_rental_qty	decimal	9	10	Y
r101_flow	net_list_rental_qty	decimal	9	10	Y
r101_flow	package_identifier	char	1	1	Y
r101_flow	key_descr2	varchar	50	50	Y
r101_flow	donees_todate	numeric	9	18	Y
r101_flow	total_orders	numeric	9	18	Y
r101_flow	tot_order_pct_rspnce	numeric	9	18	Y
r101_flow	renew_subs_pct	numeric	9	18	Y
r101_flow	import_file_type	char	1	1	Y
r101_flow	fulfillment_house	varchar	50	50	Y
r101_topline	Product_id	numeric	9	18	Y
r101_topline	src_cat_name	varchar	30	30	Y
r101_topline	source_name	varchar	250	250	Y
r101_topline	camp_name	varchar	250	250	Y
r101_topline	gross_subs	numeric	9	18	Y
r101_topline	tot_net_subs	numeric	9	18	Y



## IMT Subscription Marketing Reporting System

Table Name	Field Name	Type	Length	Precision	Nullable
r101_toplevel	camp_date_fr	datetime	8	23	Y
r101_toplevel	Budgeted_gross_pct	numeric	9	18	Y
r101_toplevel	Budgeted_vol	numeric	9	18	Y
r101_toplevel	promo_qty	numeric	9	18	Y
r401_flow	r401_rec_num	numeric	9	18	N
r401_flow	r401_reorg_date	smalldatetime	4	16	N
r401_flow	product_id	numeric	5	8	N
r401_flow	r401_key	varchar	25	25	N
r401_flow	source_identifier	char	1	1	Y
r401_flow	credit_period_identifier	varchar	6	6	Y
r401_flow	current_source_identifier	varchar	5	5	Y
r401_flow	effort_identifier	char	1	1	Y
r401_flow	split_identifier	varchar	10	10	Y
r401_flow	select_date	smalldatetime	4	16	Y
r401_flow	bills_mailed	numeric	9	12	Y
r401_flow	bills_mailed_amt	numeric	9	12	Y
r401_flow	this_week_pmt	numeric	9	12	Y
r401_flow	total_pmt	numeric	9	12	Y
r401_flow	pmt_pct	numeric	5	5	Y
r401_flow	step_up	int	4	10	Y
r401_flow	step_up_pct	numeric	5	5	Y
r401_flow	pmt_plus_step_up_rev	numeric	9	12	Y
r401_flow	avg_rev_per_pmt	numeric	5	6	Y
r401_flow	avg_term_per_pmt	numeric	5	6	Y
r401_flow	request_cancel	numeric	9	12	Y
r401_flow	request_cancel_pct	numeric	5	5	Y
r401_flow	per_cancel	numeric	9	12	Y
r401_flow	per_cancel_pct	numeric	5	5	Y
r401_flow	pmt_on_pmt	numeric	9	12	Y
r401_flow	pmt_after_cancel	numeric	9	12	Y
r401_flow	total_copies	numeric	9	12	Y
r401_flow	fulfillment_title_name	varchar	50	50	Y
r401_flow	est_max_eff_mail_qty	numeric	9	12	Y
r401_flow	import_file_type	char	1	1	Y
r401_summary	r401_rec_num	numeric	9	18	N
r401_summary	r401_reorg_date	smalldatetime	4	16	N
r401_summary	product_id	numeric	5	8	N
r401_summary	r401_key	varchar	25	25	N
r401_summary	source_identifier	char	1	1	Y
r401_summary	credit_period_identifier	varchar	6	6	Y
r401_summary	current_source_identifier	varchar	5	5	Y
r401_summary	effort_identifier	char	1	1	Y
r401_summary	split_identifier	varchar	10	10	Y
r401_summary	select_date	smalldatetime	4	16	Y
r401_summary	bills_mailed	numeric	9	12	Y
r401_summary	bills_mailed_amt	numeric	9	12	Y
r401_summary	this_week_pmt	numeric	9	12	Y



Table Name	Field Name	Type	Length	Precision	Nullable
r401_summary	total_pmt	numeric	9	12	Y
r401_summary	pmt_pct	numeric	5	5	Y
r401_summary	step_up	int	4	10	Y
r401_summary	step_up_pct	numeric	5	5	Y
r401_summary	pmt_plus_step_up_rev	numeric	9	12	Y
r401_summary	avg_rev_per_pmt	numeric	5	6	Y
r401_summary	avg_term_per_pmt	numeric	5	6	Y
r401_summary	request_cancel	numeric	9	12	Y
r401_summary	request_cancel_pct	numeric	5	5	Y
r401_summary	per_cancel	numeric	9	12	Y
r401_summary	per_cancel_pct	numeric	5	5	Y
r401_summary	pmt_on_pmt	numeric	9	12	Y
r401_summary	pmt_after_cancel	numeric	9	12	Y
r401_summary	total_copies	numeric	9	12	Y
r401_summary	fulfillment_title_name	varchar	50	50	Y
r401_summary	est_max_eff_mail_qty	numeric	9	12	Y
r401_summary	import_file_type	varchar	50	50	Y
rankings_base_info	product_id	numeric	5	8	N
rankings_base_info	src_cat_name	varchar	30	30	N
rankings_base_info	source_name	varchar	250	250	N
rankings_base_info	camp_name	varchar	250	250	N
rankings_base_info	panel_name	varchar	250	250	N
rankings_base_info	gross_subs_pct	numeric	5	7	Y
rankings_base_info	net_subs_pct	numeric	5	7	Y
rankings_base_info	net_per_net_sub	money	8	19	Y
revenue_sets	product_id	numeric	9	18	N
revenue_sets	r_set_name	varchar	50	50	N
revenue_sets	subs_revenue	numeric	9	11	Y
revenue_sets	subs_cpm_cpu	varchar	10	10	Y
revenue_sets	ad_revenue	numeric	9	11	Y
revenue_sets	ad_cpm_cpu	varchar	10	10	Y
revenue_sets	listrental_revenue	numeric	9	11	Y
revenue_sets	listrental_cpm_cpu	varchar	10	10	Y
revenue_sets	other_revenue	numeric	9	11	Y
revenue_sets	other_cpm_cpu	varchar	10	10	Y
revenue_sets	rec_id	numeric	9	18	N
rpt_cats	report_id	numeric	9	18	N
rpt_cats	src_cat_name	varchar	30	30	N
rpt_cats	src_sub_cat	varchar	50	50	Y
rpt_cats	location	varchar	250	250	N
rpt_cats	rpt_name	varchar	250	250	N
rpt_cats	rpt_description	varchar	250	250	Y
rpt_cats	rpt_filters	varchar	250	250	Y
rpt_defs	rpt_id	numeric	9	18	N
rpt_defs	product_id	numeric	9	18	Y
rpt_defs	rpt_group	varchar	75	75	N
rpt_defs	rpt_title	varchar	75	75	N



## IMT Subscription Marketing Reporting System

Table Name	Field Name	Type	Length	Precision	Nullable
rpt_defs	rpt_description	varchar	250	250	Y
rpt_defs	rpt_selFormula	varchar	1000	1000	N
rpt_defs	rpt_path	varchar	250	250	N
rpt_defs	rpt_param1	varchar	50	50	Y
rpt_defs	rpt_param2	varchar	50	50	Y
source	src_cat_name	varchar	30	30	N
source	source_name	varchar	250	250	N
source	product_id	numeric	9	18	N
source	src_indicator	char	3	3	N
source	grp_name	varchar	50	50	Y
source	title_name	varchar	50	50	Y
source	rec_id	numeric	9	18	N
source_descriptions	u_source_name	varchar	250	250	N
source_descriptions	u_source_description	varchar	250	250	Y
source_descriptions	rec_id	numeric	9	18	N
src_category	product_id	numeric	9	18	N
src_category	src_cat_name	varchar	30	30	N
src_category	src_cat_descr	varchar	250	250	Y
src_category	grp_name	varchar	50	50	Y
src_category	title_name	varchar	50	50	Y
src_category	rec_id	numeric	9	18	N
src_channel	src_channel_name	varchar	30	30	N
src_channel	src_channel_descr	varchar	250	250	Y
starts	product_id	numeric	9	18	N
starts	full_promo_key	varchar	25	25	N
starts	start_id	varchar	25	25	N
starts	term	numeric	9	18	N
starts	grs_revenue	money	8	19	Y
starts	net_revenue	money	8	19	Y
starts	grs_subs	numeric	9	18	Y
starts	net_subs	numeric	9	18	Y
starts	cancels	numeric	9	18	Y
starts	reinstates	numeric	9	18	Y
starts	start_reorg_date	smalldatetime	4	16	N
starts	src_indicator	char	3	3	Y
starts	rec_id	char	10	10	Y
starts	import_file_type	char	1	1	Y
Summary_log	summary_date	datetime	8	23	Y
Summary_log	descr	varchar	50	50	Y
Summary_log	rec_id	int	4	10	N



## ASP PAGES

### OPEN PAGES

Welcome.asp  
Adminindex.asp  
Adminindexlogin.asp  
Noadminrights.asp  
Nosetuprights.asp

### MAIN ADMIN PAGES

Admin1a.asp  
Admin2a.asp  
Admin3a.asp  
Admin4a.asp  
Admin5a.asp  
Admin6a.asp  
Adminkeysleft.asp  
Adminkeysright.asp  
Addsource.asp  
Editsource.asp  
Ms\_editcampaign.asp  
Ms\_editpanel.asp  
Ms\_editkey.asp

### TOOL BAR

Drill.asp  
Drillbounce.asp  
Adminsearch.asp  
Searchbounce.asp  
Avail.asp  
Allsetuptab.asp  
Setuptabbounce.asp

### SETUP (LOOKUP) TABLES

Tabautosetup.asp  
Edit\_autosetup.asp  
Edit\_campaigndef.asp  
Edit\_renewal\_psdef.asp  
Edit\_panel\_def.asp  
Setupcampstat.asp  
Editcampstat.asp  
Addcampstat.asp  
Setupcamptype.asp  
Addcamptype.asp  
Editcamptype.asp  
Setupeset  
Addeset.asp  
Editeset.asp

Setuplistcat.asp  
Editlistcat.asp  
Addlistcat.asp  
Setuplistname.asp  
Editlistname.asp  
Addlistname.asp  
Setuplistsegment.asp  
Editlistsegment.asp  
Addlistsegment.asp  
Setuppanelpackage.asp  
Editpanelpackage.asp  
Addpanelpackage.asp  
Setuppanelsubtype.asp  
Editpanelsubtype.asp  
Addpanelsubtype.asp  
Setuppaneltesttype.asp  
Editpaneltesttype.asp  
Addpaneltesttype.asp  
Setuppaneltype.asp  
Editpaneltype.asp  
Addpaneltype.asp  
Setuprset.asp  
Editrset.asp  
Addrset.asp  
Setupsourcename.asp  
Editsourcename.asp  
Addsourcename.asp  
Setuppriorsrcdesc.asp  
Editpriorsrcdesc.asp  
Addpriorsrcdesc.asp

### REPORTING

Login.asp  
Choosecat.asp  
Choosecat2.asp  
Chooserppt.asp  
Reporta.asp  
Reportb.asp  
Reportc.asp  
Reportd.asp  
Reporte.asp  
Reportf.asp  
Reportg.asp  
Saverpt.asp  
Choosesaved.asp  
Choosesavedb.asp  
Choosesavedc.asp  
Choosesavedd.asp



ASP PAGE CODE  
(pages are listed in alpha order)



Add\_autosetup.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%
if Request.form("IsLoaded") = "true" then
    l_product_id = Session("PRODUCT_ID")
    l_src_indicator = Request.form("P_SRC_INDICATOR")
    l_source_name = Request.form("P_SOURCE_NAME")
    if l_source_name = "0" then l_source_name = "Undefined"
    objRST.open "select src_cat_name from source where source_name = " & l_source_name & "", strSQLOLEDB,
adOpenStatic, adLockOptimistic
    if not objRST.eof then l_src_cat_name = objRST(0) else l_src_cat_name = "Undefined"
    objRST.close
    l_valid_from = Request.form("P_VALID_FROM")
    l_valid_to = Request.Form("P_VALID_TO")

    l_setup_mask = Request.Form("P_SETUP_MASK")
    l_e_set_name = Request.Form("P_E_SET_NAME")
    if l_e_set_name = "0" then l_e_set_name = null

    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_add_autosetup_scheme"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    objCMD.Parameters(1).Value = l_product_id
    objCMD.Parameters(2).Value = l_src_cat_name
    objCMD.Parameters(3).Value = l_src_indicator
    objCMD.Parameters(4).Value = l_source_name
    objCMD.Parameters(5).Value = l_valid_from
    objCMD.Parameters(6).Value = l_valid_to
    objCMD.Parameters(7).Value = l_setup_mask
    objCMD.Parameters(8).Value = l_e_set_name
    Set TempNonSessionVar = objCMD.Execute()
    response.redirect ("tabautosetup.asp")
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="add_autosetup.asp" name=form1 id=form1>
<table width="640" border="0" cellpadding="2">
<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<p><noabr>

<input type="image" src="../graphics/submit1222.gif" width="54" height="16" border="0" alt="Submit to Database">

```

```

<font face="Verdana, Arial, Helvetica, sans-serif" size="-4"><a href="tabautosetup.asp">Return
to AutoSetup Parameters Menu</a></font></nobr> </p>
<table width="265" border="0" cellspacing="3" cellpadding="2">
<tr>
<td bgcolor="#003366" height="25" width="166" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">SOURCE
NAME </font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<%
strSQL = "select U_SOURCE_NAME FROM Source_descriptions "
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SOURCE_NAME",6, "SELECT SOURCE",
""
objRST.close
%>
</font></b></font></div>
</td>
<tr>
<td bgcolor="#003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font
size="1">SOURCE<br>
INDICATOR </font></b></font>
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_SRC_INDICATOR' id=0 maxlength="30" size="6">
</font></b></font>
</div>
</td>
<tr>
<td bgcolor="#003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">VALID
FROM </font></b></font>
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_VALID_FROM' id=0 maxlength="30" size="15">
</font></b></font>
</div>
</td>
<tr>
<td bgcolor="#003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">VALID TO
</font></b></font>
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_VALID_TO' id=0 maxlength="30" size="15">
</font></b></font>
</div>
</td>
</tr>

```



```
<tr>
  <td bgcolor="003366" height="25" width="41" align="center">
    <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">SETUP
MASK </font></b></font>
    </div>
  </td>
  <td bgcolor="#99CCFF" height="20" width="360" align="left">
    <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
      <input type="TEXT" NAME="P_SETUP_MASK" id=0 maxlength="30" size="15">
      </font></b></font>
    </div>
  </td>
</tr>

<tr>
  <td bgcolor="003366" height="25" width="41" align="center">
    <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">EXPENSE
SET NAME</font></b></font>
    </div>
  </td>
  <td bgcolor="#99CCFF" height="20" width="360" align="left">
    <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
      <%
        strSQL = "select e_set_name FROM expense_sets where product_id = " &
Session("PRODUCT_ID") & " order by e_set_name "
        objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        showQueryListBox objRST, "P_E_SET_NAME",6, "SELECT EXPENSE
SET", l_e_set_name
        objRST.close %>
      </font></b></font>
    </div>
  </td>
</tr>
</table>
<p>&nbsp;</p>

</td>
</tr>
</table>
<input type="HIDDEN" value="true" name="IsLoaded">
</form>
</body>
</html>
```

add\_campaign\_def.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%
if Request.form("IsLoaded") = "true" then
    l_product_id = Session("PRODUCT_ID")
    l_campaign_code = request.form("P_CAMPAIGN_CODE")
    l_campaign_description = request.form("P_CAMPAIGN_DESCRIPTION")
    l_src_cat_name = Request.form("P_SRC_CAT_NAME")
    if not(l_src_cat_name > "0") then l_src_cat_name = ""
    l_campaign_date = request.form("P_CAMPAIGN_DATE")
    if not(l_campaign_date > " ") then l_campaign_date = null
    l_panel_scheme = Request.Form("P_PANEL_SCHEME")
    if not(l_panel_scheme > "0") then l_panel_scheme = ""

    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_add_campaign_def"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    objCMD.Parameters(1).Value = l_product_id
    objCMD.Parameters(2).Value = l_campaign_code
    objCMD.Parameters(3).Value = l_campaign_description
    objCMD.Parameters(4).Value = l_src_cat_name
    objCMD.Parameters(5).Value = l_campaign_date
    objCMD.Parameters(6).Value = l_panel_scheme

    Set TempNonSessionVar = objCMD.Execute()
    response.redirect ("tabautosetup.asp")
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="add_campaign_def.asp" name=form1 id=form1>
<table width="640" border="0" cellpadding="2">
<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<p><noabr>

<input type="image" src="../graphics/submit1222.gif" width="54" height="16" border="0" alt="Submit to Database">
<font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="tabautosetup.asp">Return
to AutoSetup Parameters Menu</a></font></noabr> </p>
<td width="265" border="0" cellspacing="3" cellpadding="2">
<tr>
<td bgcolor="003366" height="25" width="166" align="center">
```



```
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">CAMPAIGN
CODE </font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_CAMPAIGN_CODE' id=0 maxlength="30" size="6">
</font></b></font></div>
</td>
</tr>
<tr>
<td bgcolor="#003366" height="25" width="166" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">CAMPAIGN
DESCRIPTION </font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_CAMPAIGN_DESCRIPTION' id=0 maxlength="50"
size="50">
</font></b></font></div>
</td>
</tr>
<tr>
<td bgcolor="#003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font
size="1">SOURCE<br>
CATEGORY </font></b></font>
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<% strSQL = "select distinct src_cat_name from source where product_id = " &
Session("PRODUCT_ID") & " order by src_cat_name"
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SRC_CAT_NAME", 6, "SELECT SOURCE
CATEGORY", l_src_cat_name
objRST.close
'<input type="TEXT" NAME='P_SRC_IDENTIFIER' id=0 maxlength="30"
size="6">
%>
</font></b></font>
</div>
</td>
</tr>
<tr>
<td bgcolor="#003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">CAMPAIGN
DATE </font></b></font>
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_CAMPAIGN_DATE' id=0 maxlength="30" size="6">
</font></b></font>
</div>
</td>
</tr>
<tr>
<td bgcolor="#003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">PANEL
SCHEME </font></b></font>
```



```
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
  <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
    <% strSQL = "select distinct panel_scheme from autosetup_panel_def where product_id
= " & Session("PRODUCT_ID") & " order by panel_scheme"
    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    showQueryListBox objRST, "P_PANEL_SCHEME", 6, "SELECT PANEL
SCHEME", l_panel_scheme
    objRST.close
    ' <input type="TEXT" NAME="P_SRC_IDENTIFIER" id=0 maxlength="30"
size="6">
    %>
    </font></b></font>
  </div>
</td>
</tr>
</table>
<input type="HIDDEN" value="true" name="IsLoaded">
</form>
</body>
</html>
```

add\_panel\_def.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%
if Request.form("IsLoaded") = "true" then
    l_product_id = Session("PRODUCT_ID")
    l_panel_scheme = request.form("P_PANEL_SCHEME")
    l_panel_codes = request.form("P_PANEL_CODES")
    l_panel_name = Request.form("P_PANEL_NAME")

    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_add_panel_def"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    objCMD.Parameters(1).Value = l_product_id
    objCMD.Parameters(2).Value = l_panel_scheme
    objCMD.Parameters(3).Value = l_panel_codes
    objCMD.Parameters(4).Value = l_panel_name

    Set TempNonSessionVar = objCMD.Execute()
    response.redirect ("tabautosetup.asp")
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="add_panel_def.asp" name=form1 id=form1>
<table width="640" border="0" cellpadding="2">
<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<p><nobr>

<input type="image" src="../graphics/submit1222.gif" width="54" height="16" border="0" alt="Submit to Database">
<font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="tabautosetup.asp">Return
to AutoSetup Parameters Menu</a></font></nobr> </p>
<table width="480" border="0" cellspacing="3" cellpadding="2">
<tr>
<td bgcolor="003366" height="25" width="166" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">PANEL
SCHEME </font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME="P_PANEL_SCHEME" id=0 maxlength="50" size="50">

```



```

        </font></b></font></div>
    </td>
</tr>

<tr>
<td bgcolor="003366" height="25" width="166" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">PANEL
CODES </font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_PANEL_CODES' id=0 maxlength="250" size="100">
</font></b></font></div>
</td>
</tr>

<tr>
<td bgcolor="003366" height="25" width="166" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">PANEL
NAME </font></b></font>
</div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="TEXT" NAME='P_PANEL_NAME' id=0 maxlength="50" size="50">
</font></b></font>
</div>
</td>
</tr>
</table>
<input type="HIDDEN" value="true" name="IsLoaded">
</form>
</body>
</html>
```



add\_renewal\_psdef.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%
if Request.form("IsLoaded") = "true" then
    l_product_id = Session("PRODUCT_ID")
    l_src_indicator = Request.form("P_SRC_INDICATOR")
    l_prior_src_indicator = Request.form("P_PRIOR_SRC_INDICATOR")
    l_source_name = Request.form("P_SOURCE_NAME")
    objRST.open "select src_cat_name from source where source_name = "" & l_source_name & """, strSQLOLEDB,
adOpenStatic, adLockOptimistic
    if not objRST.eof then l_src_cat_name = objRST(0) else l_src_cat_name = "Undefined"
    objRST.close
    l_valid_from = Request.form("P_VALID_FROM")
    l_valid_to = Request.Form("P_VALID_TO")
    response.write (l_prior_src_indicator)
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_add_renewal_psdef"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    objCMD.Parameters(1).Value = l_product_id
    objCMD.Parameters(2).Value = l_src_cat_name
    objCMD.Parameters(3).Value = l_src_indicator
    objCMD.Parameters(4).Value = l_prior_src_indicator
    objCMD.Parameters(5).Value = l_source_name
    objCMD.Parameters(6).Value = l_valid_from
    objCMD.Parameters(7).Value = l_valid_to
    Set TempNonSessionVar = objCMD.Execute()
    response.redirect ("tabautosetup.asp")
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="add_renewal_psdef.asp" name=form1 id=form1>
<table width="640" border="0" cellpadding="2">
<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<p><noabr>

<input type="image" src="../graphics/submit1222.gif" width="54" height="16" border="0" alt="Submit to Database">
<font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="tabautosetup.asp">Return
to AutoSetup Parameters Menu</a></font></noabr> </p>
<table width="265" border="0" cellspacing="3" cellpadding="2">
<tr>
```

```

<td bgcolor="003366" height="25" width="166" align="center">
  <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">SOURCE
    NAME </font></b></font></div>
</td>
      <td bgcolor="#99CCFF" height="20" width="360" align="left">
        <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
          <%
            strSQL = "select U_SOURCE_NAME FROM Source_descriptions "
            objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
            showQueryListBox objRST, "P_SOURCE_NAME",6, "SELECT SOURCE",
**
            objRST.close
          %>
        </font></b></font></div>
      </td>
    </tr>
  <tr>
    <td>
      <td bgcolor="003366" height="25" width="41" align="center">
        <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font
size="1">SOURCE<br>
        INDICATOR </font></b></font>
        </div>
      </td>
      <td bgcolor="#99CCFF" height="20" width="360" align="left">
        <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
          <input type="TEXT" NAME="P_SRC_INDICATOR" id=0 maxlength="30" size="6">
          </font></b></font>
        </div>
      </td>
    </tr>
  <tr>
    <td>
      <td bgcolor="003366" height="25" width="41" align="center">
        <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">PRIOR
SOURCE<br>
        INDICATOR </font></b></font>
        </div>
      </td>
      <td bgcolor="#99CCFF" height="20" width="360" align="left">
        <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
          <input type="TEXT" NAME="P_PRIOR_SRC_INDICATOR" id=0 maxlength="250"
size="100">
          </font></b></font>
        </div>
      </td>
    </tr>
  <tr>
    <td>
      <td bgcolor="003366" height="25" width="41" align="center">
        <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">VALID
FROM </font></b></font>
        </div>
      </td>
      <td bgcolor="#99CCFF" height="20" width="360" align="left">
        <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
          <input type="TEXT" NAME="P_VALID_FROM" id=0 maxlength="30" size="15">
          </font></b></font>
        </div>
      </td>
    </tr>
  <tr>
    <td>
      <td bgcolor="003366" height="25" width="41" align="center">

```

```
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">VALID TO
</font></b></font>
    </div>
    <td>
        <td bgcolor="#99CCFF" height="20" width="360" align="left">
            <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
                <input type="TEXT" NAME='P_VALID_TO' id=0 maxlength="30" size="15">
                </font></b></font>
            </div>
        </td>
    </tr>
</table>
<p>&nbsp;</p>
</td>
</tr>
</table>
<input type="HIDDEN" value="true" name="IsLoaded">
</form>
</body>
</html>.
```

addcamp.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%

'on error resume next

dim p_camp(4), p_camp_type(4), p_camp_status(4), p_camp_date_fr(4)
dim p_camp_date_to(4), p_agt_code(4), p_agt_descr(4)
dim p_e_set_name(4), p_r_set_name(4), p_newstand_sale(4)
dim p_budgeted_gross_pct(4), p_budgeted_vol(4)
ADDED = FALSE
errors = false
' Response.Write "This should work"
for i = 0 to 4
    p_camp(i) = REQUEST("p_camp" & cstr(i))
    p_camp_type(i) = REQUEST("p_camp_type" & cstr(i))
    p_camp_status(i) = REQUEST("p_camp_status" & cstr(i))
    p_camp_date_fr(i) = REQUEST("p_camp_date_fr" & cstr(i))
    p_camp_date_to(i) = REQUEST("p_camp_date_to" & cstr(i))
    p_agt_code(i) = REQUEST("p_agt_code" & cstr(i))
    p_agt_descr(i) = REQUEST("p_agt_descr" & cstr(i))
    p_e_set_name(i) = REQUEST("p_e_set_name" & cstr(i))
    p_r_set_name(i) = REQUEST("p_r_set_name" & cstr(i))
    p_newstand_sale(i) = REQUEST("p_newstand_sale" & cstr(i))
    p_budgeted_gross_pct(i) = REQUEST("p_budgeted_gross_pct" & cstr(i))
    p_budgeted_vol(i) = REQUEST("p_budgeted_vol" & cstr(i))
    'Response.Write p_camp(i) & "<br>" & p_camp_status(i)
    ' Response.Write p_source(i) & p_src_ind(i)
    if p_camp(i) > " " then
        added = true
        ' Response.Write p_camp(i) & "<br>" & p_camp_status(i)
        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_Add_Campaign"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = p_camp(i)
        objCMD.Parameters(2).Value = ValidatePickList(p_camp_type(i))
        objCMD.Parameters(3).Value = ValidatePickList(p_camp_status(i))
        objCMD.Parameters(4).Value = validateDate(p_camp_date_fr(i))
        objCMD.Parameters(5).Value = null
        objCMD.Parameters(6).Value = null
        objCMD.Parameters(7).Value = null
        objCMD.Parameters(8).Value = ValidatePickList(p_e_set_name(i))
        objCMD.Parameters(9).Value = ValidatePickList(p_r_set_name(i))
        objCMD.Parameters(10).Value = null
        objCMD.Parameters(11).Value = ValidateNumber(p_newstand_sale(i))
        objCMD.Parameters(12).Value = ValidateNumber(p_budgeted_gross_pct(i))
        objCMD.Parameters(13).Value = ValidateNumber(p_budgeted_vol(i))

        objCMD.Parameters(13).Value = Session("SOURCE_NAME")
        objCMD.Parameters(14).Value = Session("SRC_INDICATOR")
        objCMD.Parameters(15).Value = Session("PRODUCT_ID")
        objCMD.Parameters(16).Value = Session("SRC_CAT_NAME")
        objCMD.Parameters(17).Value = Session("GRP_NAME")
        objCMD.Parameters(18).Value = Session("TITLE_NAME")
        Set TempNonSession Var = objCMD.Execute()
        ' If objCMD.Errors.Count > 0 Then
        '     Response.Write " DAMN YOU KEVIN <BR>" & objCMD.Errors.Count
        '     Response.Write " errors " & objCMD.Errors.Count & "<BR>"
        '     errStr = errStr & "Campaign <b>" & p_camp(i) & " not added <br>"
        '     errors = true
        ' end if

```



```
end if
next
if added = true then 'and errors = false then
Response.Redirect("admin5a.asp")
END IF
%>
<html>
<head>
<title>addCamp.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="addCamp.asp" name=form1 id=form1>
<table width="640" border="0" cellpadding="2">

<% prevSelections (4) %>
<% if errors = true then %>
<tr valign="top">
<td>
The following Errors Occured: <br>
<% Response.Write errStr %>
<br>
<a href="addCamp.asp"> Add Campaign Names</a><br>
<a href="admin5a.asp"> Return to Main Campaign Screen</a><br>

<% else %>
<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<p><no>


<input type="image" src="..graphics/submit1222.gif" width="54" height="16" border="0" alt="Submit to Database" id=image1
name=image1>
<font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="admin5a.asp">Return
to Campaign</a></font></no> </p>
<table border="0" cellspacing="3" cellpadding="2">
<tr>
<td bgcolor="003366" height="25" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">
CAMPAIGN NAME
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
CAMPAIGN START DATE
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
CAMPAIGN TYPE
</font></b></font></div></td>
```

```

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
CAMPAIGN STATUS
</font></b></font></div></td>

```

```

<%
    <td bgcolor="003366" height="25" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    DATE TO
    </font></b></font></div></td>
    <td bgcolor="003366" height="25" width="41" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    AGENT CODE
    </font></b></font></div></td>
    <td bgcolor="003366" height="25" width="41" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    AGENT DESCR
    </font></b></font></div></td> %>
    <td bgcolor="003366" height="25" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    EXPENSE SET
    </font></b></font></div></td>

    <td bgcolor="003366" height="25" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    REVENUE SET
    </font></b></font></div></td>
    <%> <td bgcolor="003366" height="25" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    NEWSSTAND SALE
    </font></b></font></div></td>
    %>
    <td bgcolor="003366" height="25" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    BUDGETED GROSS %
    </font></b></font></div></td>

    <td bgcolor="003366" height="25" align="center">
    <div align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
    <b><font size="1">
    BUGETED VOLUME
    </font></b></font></div></td>
    </tr>

<% for i = 0 to 4 %>
<tr>
<td height="15" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

```

```



```



```
<input type="text" name="p_budgeted_gross_pct<%Response.write i %>" maxlength="30" size="8">
</font></b></font></div>
</td>
<td height="15" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<input type="text" name="p_budgeted_vol<%Response.write i %>" maxlength="30" size="8">
</font></b></font></div>
</td>
</tr>

<% next %>

</table>
<p>&nbsp;</p>
<% end if %>
</td>
</tr>
</table>
</form>
</body>
</html>
```



addcampstat.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addCampStat.asp"
l_col_names(0) = "CAMPAIGN STATUS"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Prod_CAMP_STATUS"
l_Title = " Campaign Status "
l_setupTabLink = "setupCampStat.asp"
l_showPrevSelection= true

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names , l_ColumnCount
addBottom %>
```

addcamptype.asp

```
<%@ Language=VBScript %>
<!-- #include file=../support/global_constants.asp -->
<!-- #include file=../support/dbConnection.asp -->
<!-- #include file=../support/adoUtils.asp -->
<!-- #include file=../support/srcTab.asp -->
<!-- #include file=../support/setupTab.asp -->
<%

dim l_col_names(2)
l_file = "addCampType.asp"
l_col_names(0) = "CAMPAIGN TYPE"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Prod_CAMP_Type"
l_Title = " Campaign Type "
l_setupTabLink = "setupCampType.asp"
l_showPrevSelection= true

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names , l_ColumnCount
addBottom %>
```

addeset.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->
<!-- #include file="..\support/PickListObj.asp" -->
<%
```

```
If Request.Form("IsLoaded") <> "" then
```

```
    product_id = Session("PRODUCT_ID")
    e_set_name = Request.form("e_set_name")
    print_cost = Request.form("print_cost")
    print_cpm_cpu = Request.form("print_cpm_cpu")
    print_basis = Request.form("print_basis")
    merge_purge = Request.form("merge_purge")
    merge_cpm_cpu = Request.form("merge_cpm_cpu")
    merge_basis = Request.form("merge_basis")
    ltr_cost = Request.form("ltr_cost")
    ltr_cpm_cpu = Request.form("ltr_cpm_cpu")
    ltr_basis = Request.form("ltr_basis")
    postage_out_cost = request.form("postage_out_cost")
    postage_out_cpm_cpu = Request.form("postage_out_cpm_cpu")
    postage_out_basis = Request.form("postage_out_basis")
    postage_in_cost = Request.form("postage_in_cost")
    postage_in_cpm_cpu = Request.form("postage_in_cpm_cpu")
    postage_in_basis = Request.form("postage_in_basis")
    premium_cost = Request.form("premium_cost")
    premium_cpm_cpu = Request.form("premium_cpm_cpu")
    premium_basis = Request.form("premium_basis")
    badpay_cost = Request.form("badpay_cost")
    badpay_cpm_cpu = Request.form("badpay_cpm_cpu")
    badpay_basis = Request.form("badpay_basis")
    billing_cost = Request.form("billing_cost")
    billing_cpm_cpu = Request.form("billing_cpm_cpu")
    billing_basis = Request.form("billing_basis")
    subs_svc_costs = Request.form("subs_svc_costs")
    subs_svc_cpm_cpu = Request.form("subs_svc_cpm_cpu")
    subs_svc_basis = Request.form("subs_svc_basis")
    exp_other_costs = Request.form("exp_other_costs")
    exp_other_cpm_cpu = Request.form("exp_other_cpm_cpu")
    exp_other_basis = Request.form("exp_other_basis")
```

```
objCMD.ActiveConnection = strSQLOLEDB
objCMD.CommandText = "sp_add_prod_e_set"
objCMD.CommandType = adCmdStoredProc
objCMD.Parameters.Refresh
    if not isnumeric(product_id) then
        response.redirect("setupeset.asp")
    else
        objCMD.Parameters("@product_id").Value = product_id
    end if
    objCMD.Parameters("@e_set_name").Value = e_set_name
    if not isnumeric(print_cost) then
        objCMD.Parameters("@print_cost").Value = null
    else
        objCMD.Parameters("@print_cost").Value = print_cost
    end if
    objCMD.Parameters("@print_cpm_cpu").Value = print_cpm_cpu
    objCMD.Parameters("@print_basis").Value = print_basis
    if not isnumeric(merge_purge) then
        objCMD.Parameters("@merge_purge").Value = null
    else
        objCMD.Parameters("@merge_purge").Value = merge_purge
    end if
```

```

objCMD.Parameters("@merge_cpm_cpu").Value = merge_cpm_cpu
objCMD.Parameters("@merge_basis").Value = merge_basis
if not (ltr_cost > " ") then
    objCMD.Parameters("@ltr_cost").Value = null
else
    objCMD.Parameters("@ltr_cost").Value = ltr_cost
end if
objCMD.Parameters("@ltr_cpm_cpu").Value = ltr_cpm_cpu
objCMD.Parameters("@ltr_basis").Value = ltr_basis
if not (postage_out_cost > " ") then
    objCMD.Parameters("@postage_out_cost").value = null
else
    objCMD.Parameters("@postage_out_cost").Value = postage_out_cost
end if
objCMD.Parameters("@postage_out_cpm_cpu").Value = postage_out_cpm_cpu
objCMD.Parameters("@postage_out_basis").Value = postage_out_basis
if not (postage_in_cost > " ") then
    objCMD.Parameters("@postage_in_cost") = null
else
    objCMD.Parameters("@postage_in_cost").Value = postage_in_cost
end if
objCMD.Parameters("@postage_in_cpm_cpu").Value = postage_in_cpm_cpu
objCMD.Parameters("@postage_in_basis").Value = postage_in_basis
if not (premium_cost > " ") then
    objCMD.Parameters("@premium_cost") = null
else
    objCMD.Parameters("@premium_cost").Value = premium_cost
end if
objCMD.Parameters("@premium_cpm_cpu").Value = premium_cpm_cpu
objCMD.Parameters("@premium_basis").Value = premium_basis
if not (badpay_cost > " ") then
    objCMD.Parameters("@badpay_cost").value = null
else
    objCMD.Parameters("@badpay_cost").Value = badpay_cost
end if
objCMD.Parameters("@badpay_cpm_cpu").Value = badpay_cpm_cpu
objCMD.Parameters("@badpay_basis").Value = badpay_basis
if not (billing_cost > " ") then
    objCMD.Parameters("@billing_cost").Value = null
else
    objCMD.Parameters("@billing_cost").Value = billing_cost
end if
objCMD.Parameters("@billing_cpm_cpu").Value = billing_cpm_cpu
objCMD.Parameters("@billing_basis").Value = billing_basis
if not (subs_svc_costs > " ") then
    objCMD.Parameters("@subs_svc_costs").value = null
else
    objCMD.Parameters("@subs_svc_costs").Value = subs_svc_costs
end if
objCMD.Parameters("@subs_svc_cpm_cpu").Value = subs_svc_cpm_cpu
objCMD.Parameters("@subs_svc_basis").Value = subs_svc_basis
if not (exp_other_costs > " ") then
    objCMD.Parameters("@exp_other_costs") = null
else
    objCMD.Parameters("@exp_other_costs").Value = exp_other_costs
end if
objCMD.Parameters("@exp_other_cpm_cpu").Value = exp_other_cpm_cpu
objCMD.Parameters("@exp_other_basis").Value = exp_other_basis
Set TempNonSessionVar = objCMD.Execute()
Response.Redirect "setupeset.asp"

end if
%>

<html>
<head>
<title>AddESet.asp</title>
</head>

```



```
<script language="JavaScript">
<!--
function MM_swapImgRestore() { //v2.0
  if (document.MM_swapImgData != null)
    for (var i=0; i<(document.MM_swapImgData.length-1); i+=2)
      document.MM_swapImgData[i].src = document.MM_swapImgData[i+1];
}

function MM_preloadImages() { //v2.0
  if (document.images) {
    var imgFiles = MM_preloadImages.arguments;
    if (document.preloadArray==null) document.preloadArray = new Array();
    var i = document.preloadArray.length;
    with (document) for (var j=0; j<imgFiles.length; j++) if (imgFiles[j].charAt(0)!="#{") {
      preloadArray[i] = new Image;
      preloadArray[i++].src = imgFiles[j];
    }
  }
}

function MM_swapImage() { //v2.0
  var i,j=0,objStr,obj.swapArray=new Array,oldArray=document.MM_swapImgData;
  for (i=0; i < (MM_swapImage.arguments.length-2); i+=3) {
    objStr = MM_swapImage.arguments[(navigator.appName == "Netscape"?i:i+1)];
    if ((objStr.indexOf('document.layers[')==0 && document.layers==null) ||
        (objStr.indexOf('document.all[') ==0 && document.all ==null))
      objStr = 'document'+objStr.substring(objStr.lastIndexOf('.'),objStr.length);
    obj = eval(objStr);
    if (obj != null) {
      swapArray[j++] = obj;
      swapArray[j++] = (oldArray==null || oldArray[j-1]!=obj)?obj.src:oldArray[j];
      obj.src = MM_swapImage.arguments[i+2];
    }
  }
  document.MM_swapImgData = swapArray; //used for restore
}
//-->
</script>
<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="addEset.asp" id=form1 name=form1>
<input type="HIDDEN" name="IsLoaded" value="TRUE">

<TABLE BORDER=0 >
  <TR>
    <td> </td>
  </tr>
  <tr>
    <td colspan=5> <font face="Verdana, Arial, Helvetica, sans-serif" size="4">
      <a href="setupEset.asp"> </a></font>
      <table width="100%" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td width="18%" valign="top"></td>
          <td width="82%" colspan="3">
            <div align="right"> </div>
            <div align="right"></div>
          </td>
        </tr>
        <tr>
          <td width="18%" valign="top">&nbsp;</td>
          <td width="82%" colspan="3">
            <div align="right"> </div>
            <div align="right">
              <p>
                <input type="image" name=SUBMIT value="SUBMIT" src="./graphics/submit1222.gif" width="54" height="16"
                border="0"
                alt="Submit to Database" id=image1>
              </p>
            </div>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</table>


```

```

<p><font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="setupEset.asp">Return
to Expense Set Setup Table </a></font></p>
</div>
</td>
</tr>
</table>
<p></p>
</td>
</tr>
<tr>
<td height="10" width="140" align="center">&nbsp;</td>
<td align="left" height="10">&nbsp;</td>
<td align="left" height="10"></td>
<td align="RIGHT" height="10">&nbsp;</td>
</tr>
<tr>
<td>
Expense Set
</td>
<td>
<input type="TEXT" NAME="e_set_name" value="">
</td>
</tr>
<tr>
<td>
Print Cost
</td>
<td>
<input type="TEXT" NAME="print_cost" value="">
<br>
<% showLOVListBox "CPM,CPU","print_cpm_cpu",2,"", "" %>
<br>
<input type="HIDDEN" NAME="print_basis" value="Mail Qty">
Mail Qty
</td>
</tr>
<tr>
<td>
Merge Purge Cost
</td>
<td>
<input type="TEXT" NAME="merge_purge" value="">
<br>
<% showLOVListBox "CPM,CPU","merge_cpm_cpu",2,"", "" %>
<br>
<% showLOVListBox "Mail Qty,Merge Purge Qty","merge_basis",2,"", "" %>
</td>
</tr>
<tr>
<td>
Lettershop Cost
</td>
<td>
<input type="TEXT" NAME="ltr_cost" value="">
<br>
<% showLOVListBox "CPM,CPU","ltr_cpm_cpu",2,"", "" %>
<br>
<input type="HIDDEN" NAME="ltr_basis" value="Mail Qty">
Mail Qty
</td>
</tr>

```



	Postage Out Cost
</td>	
<td>	<input type="TEXT" NAME="postage_out_cost" value="">
</td>	
<td>	<% showLOVListBox "CPM,CPU","postage_out_cpm_cpu",2,"", "" %>
</td>	
<td>	<input type="HIDDEN" NAME="postage_out_basis" value="Mail Qty">
</td>	Mail Qty
<tr>	
<td>	Postage In Cost
</td>	
<td>	<input type="TEXT" NAME="postage_in_cost" value="">
</td>	
<td>	<% showLOVListBox "CPM,CPU","postage_in_cpm_cpu",2,"", "" %>
</td>	
<td>	<input type="HIDDEN" NAME="postage_in_basis" value="Gross Subs">
</td>	Gross Subs
</tr>	
<tr>	
<td>	Premium Cost
</td>	
<td>	<input type="TEXT" NAME="premium_cost" value="">
</td>	
<td>	<% showLOVListBox "CPM,CPU","premium_cpm_cpu",2,"", "" %>
</td>	
<td>	<% showLOVListBox "Gross Subs,Net Subs","premium_basis",2,"", "" %>
</td>	
</tr>	
<tr>	
<td>	Bad Pay Cost
</td>	
<td>	<input type="TEXT" NAME="badpay_cost" value="">
</td>	
<td>	<% showLOVListBox "CPM,CPU","badpay_cpm_cpu",2,"", "" %>
</td>	
<td>	<input type="HIDDEN" NAME="badpay_basis" value="Gross-Net Subs">
</td>	Gross-Net Subs
</tr>	
<tr>	
<td>	Billing Cost
</td>	
<td>	<input type="TEXT" NAME="billing_cost" value="">
</td>	
<td>	<% showLOVListBox "CPM,CPU","billing_cpm_cpu",2,"", "" %>
</td>	
<td>	<input type="HIDDEN" NAME="billing_basis" value="Credit Subs">



```

Credit Subs
</td>
</tr>
<tr>
<td>
Sub Service Cost
</td>
<td>
<input type="TEXT" NAME="subs_svc_costs" value="">
</td>
<td>
<% showLOVListBox "CPM,CPU","subs_svc_cpm_cpu",2,"", "" %>
</td>
<td>
<input type="HIDDEN" NAME="subs_svc_basis" value="Net Subs">
Net Subs
</td>
</tr>
<tr>
<td>
Exp Other Cost
</td>
<td>
<input type="TEXT" NAME="exp_other_costs" value="">
</td>
<td>
<% showLOVListBox "CPM,CPU","exp_other_cpm_cpu",2,"", "" %>
</td>
<td>
<% showLOVListBox "Mail Qty,Merge Purge,Gross Subs,Net
Subs","exp_other_basis" 2,"", "" %>
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```



addlistcat.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addListCat.asp"
l_col_names(0) = "LIST CATEGORY"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Prod_List_Cat"
l_Title = " List Category "
l_setupTabLink = "setupListCat.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_ColumnCount
addBottom %>
```



addlistname.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addListName.asp"
l_col_names(0) = "LIST NAME"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Prod_List_Name"
l_Title = " List Name "
l_setupTabLink = "setupListName.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, L_Col_names , l_ColumnCount
addBottom %>
```



addlistsegment.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addListSegment.asp"
l_col_names(0) = "LIST SEGMENT"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Prod_List_Segment"
l_Title = " List Segment "
l_setupTabLink = "setupListSegment.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names , l_ColumnCount
addBottom %>
```

addpanel.asp

```

<%@ Language=VBScript %>
<!-- #include file="./support/global_constants.asp" -->
<!-- #include file="./support/dbConnection.asp" -->
<!-- #include file="./support/adoUtils.asp" -->
<!-- #include file="./support/srcTab.asp" -->
<%
    l_debug = 0
    l_size = 6

    dim p_panel(6), p_panel_type(6), p_panel_sub_type(6), p_test_type(6)
    dim p_panel_offer(6), p_panel_offer_price(6), p_panel_pkg(6)
    dim p_panel_premium(6), p_panel_date_start(6), p_panel_date_end(6)
    dim p_e_set_name(6), p_r_set_name(6)
    dim p_newstand_sale_marker(6)
    'took out the redims in hopes of better performance, knock on wood

    'redim p_panel(l_size)
    'redim p_panel_type(l_size)
    'redim p_panel_sub_type(l_size)
    'redim p_test_type(l_size)
    'redim p_panel_offer(l_size)
    'redim p_panel_offer_price(l_size)
    'redim p_panel_pkg(l_size)
    'redim p_panel_premium(l_size)
    'redim p_panel_date_start(l_size)
    'redim p_panel_date_end(l_size)
    'redim p_e_set_name(l_size)
    'redim p_r_set_name(l_size)

    dim l_camp_type
    dim l_camp_status_name
    dim l_agt_code
    dim l_agt_descr
    dim l_camp_date_fr
    dim l_camp_date_to

    strSQL = "select CAMP_TYPE, CAMP_STATUS_NAME, AGT_CODE, " _
    & " AGT_DESCR, CAMP_DATE_FR, CAMP_DATE_TO, " _
    & " E_SET_NAME, R_SET_NAME " _
    & " from CAMPAIGNS where " _
    & " grp_name = " & Session("GRP_NAME") & " " _
    & " and PRODUCT_ID = " & Session("PRODUCT_ID") & " " _
    & " and src_cat_name = " & Session("SRC_CAT_NAME") & " " _
    & " and source_name = " & Session("SOURCE_NAME") & " " _
    & " and camp_name = " & Session("CAMP_NAME") & " " _
    & " and src_indicator = " & Session("SRC_INDICATOR") & " " _

    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    l_camp_type = objRST.Fields("CAMP_TYPE")
    l_camp_status_name = objRST.Fields("CAMP_STATUS_NAME")
    l_agt_code = objRST.Fields("AGT_CODE")
    l_agt_descr = objRST.Fields("AGT_DESCR")
    l_camp_date_fr = objRST.Fields("CAMP_DATE_FR")
    l_camp_date_to = objRST.Fields("CAMP_DATE_TO")
    l_e_set_name = objRST.Fields("E_SET_NAME")
    l_r_set_name = objRST.Fields("R_SET_NAME")

    objRST.close

    'Response.Write "This should work but it doesn't because I suck"
    for i = 0 to l_size - 1
        p_panel(i) = REQUEST("P_PANEL" & cstr(i))
        p_panel_type(i) = REQUEST("P_PANEL_TYPE" & cstr(i))
        p_panel_sub_type(i) = REQUEST("P_PANEL_SUB_TYPE" & cstr(i))
        p_test_type(i) = REQUEST("P_TEST_TYPE" & cstr(i))
        p_panel_offer(i) = REQUEST("P_PANEL_OFFER" & cstr(i))
        p_panel_offer_price(i) = REQUEST("P_PANEL_OFFER_PRICE" & cstr(i))
        p_panel_pkg(i) = REQUEST("P_PANEL_PKG" & cstr(i))
    
```

```

'p_panel_premium(i) = REQUEST("P_PANEL_PREMIUM" & cstr(i))
p_panel_date_start(i) = REQUEST("P_PANEL_DATE_START" & cstr(i))
p_panel_date_end(i) = REQUEST("P_PANEL_DATE_END" & cstr(i))
p_e_set_name(i) = REQUEST("P_E_SET_NAME" & cstr(i))
p_r_set_name(i) = REQUEST("P_R_SET_NAME" & cstr(i))
'p_newstand_sale_marker(i) = REQUEST("P_NEWSTAND_SALE_MARKER" & cstr(i))

' Response.Write p_camp(i) & "<br>" & p_camp_status(i)
' Response.Write p_panel(i) & "<br>" & p_panel_type(i)
  if p_panel(i) > "" then
' Response.Write p_panel(i) & "<br>" & p_panel_type(i) & p_test_type(i)
    added = true
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_Add_Panel"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    ' user supplied values
    objCMD.Parameters(1).Value = p_panel(i)
    objCMD.Parameters(2).Value = ValidatePickList(p_panel_type(i))
    objCMD.Parameters(3).Value = ValidatePickList(p_panel_sub_type(i))
    objCMD.Parameters(4).Value = ValidatePickList(p_test_type(i))
    objCMD.Parameters(5).Value = ValidatePickList(p_panel_pkg(i))
    objCMD.Parameters(6).Value = validateDate(p_panel_date_start(i))
    objCMD.Parameters(7).Value = validateDate(p_panel_date_end(i))
    objCMD.Parameters(8).Value = ValidatePickList(p_e_set_name(i))
    objCMD.Parameters(9).Value = ValidatePickList(p_r_set_name(i))
    objCMD.Parameters(10).Value = null"ValidatePickList(p_newstand_sale_marker(i))
    ' session supplied values
    objCMD.Parameters(11).Value = Session("PRODUCT_ID")
    objCMD.Parameters(12).Value = Session("GRP_NAME")
    objCMD.Parameters(13).Value = Session("TITLE_NAME")
    objCMD.Parameters(14).Value = Session("SRC_CAT_NAME")
    objCMD.Parameters(15).Value = Session("SOURCE_NAME")
    objCMD.Parameters(16).Value = Session("SRC_INDICATOR")
    objCMD.Parameters(17).Value = Session("CAMP_NAME")
    ' query supplied values
    objCMD.Parameters(18).Value = l_camp_type
    objCMD.Parameters(19).Value = l_camp_status_name
    objCMD.Parameters(20).Value = l_agt_code
    objCMD.Parameters(21).Value = l_agt_descr
    objCMD.Parameters(22).Value = l_camp_date_fr
    objCMD.Parameters(23).Value = l_camp_date_to

    Set TempNonSessionVar = objCMD.Execute()

  end if
next
if added = true then
Response.Redirect "ADMIN6A.ASP"
end if
' added April 6, 2000.
' convert Queries to LOVs to avoid running queries more than once

dim PanelTypeLOV, PanelSubTypeLOV, PanelTestTypeLOV
dim PackageLOV, ExpenseSetLOV, RevenueSetLOV

PanelTypeLOV = QueryToLOV("select PANEL_TYPE_NAME from CORP_PANEL_TYPE")
PanelSubTypeLOV = QueryToLOV("select PANEL_SUB_TYPE_NAME from CORP_PANEL_SUB_TYPE")
PanelTestTypeLOV = QueryToLOV("select PANEL_TEST_TYPE_NAME from CORP_PANEL_TEST_TYPE")
PackageLOV = QueryToLOV("select PACKAGE_NAME from PROD_PANEL_PACKAGE " &
  & "WHERE Product_id=" & Session("PRODUCT_ID"))
ExpenseSetLOV = QueryToLOV("select e_set_name from EXPENSE_SETS where " &
  & "PRODUCT_ID=" & Session("PRODUCT_ID"))
RevenueSetLOV = QueryToLOV("select r_set_name from REVENUE_SETS where " &
  & "PRODUCT_ID=" & Session("PRODUCT_ID"))
'we need the default revenue and expense sets from the campaign level

```



```
%>

<html>
<head>
<title>AddPanel.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
<form method="post" action="addPanel.asp" name=form1 id=form1>
  <table width="640" border="0" cellpadding="2">
    <%
    if l_debug = 1 then
    Response.write "l_e_set_name " & l_e_set_name
    Response.Write "<BR>"

    Response.write "l_r_set_name " & l_r_set_name
    Response.Write "<BR>"
    Response.write "PanelTypeLOV " & PanelTypeLOV
    Response.Write "<BR>"
    Response.write "PanelSubTypeLOV " & PanelSubTypeLOV
    Response.Write "<BR>"
    Response.write "PanelTestTypeLOV " & PanelTestTypeLOV
    Response.Write "<BR>"
    Response.write "PackageLOV " & PackageLOV
    Response.Write "<BR>"
    Response.write "ExpenseSetLOV " & ExpenseSetLOV
    Response.Write "<BR>"
    Response.write "RevenueSetLOV " & RevenueSetLOV
    Response.Write "<BR>"
    end if
    %>
    <%
    prevSelections (5) %>
    <% if errors = true then %>
      <tr valign="top">
        <td>
          The following Errors Occured: <br>
          <% Response.Write errStr %>
          <br>
          <a href="addCamp.asp"> Add Campaign Names</a><br>
          <a href="admin5a.asp"> Return to Main Campaign Screen</a><br>

        <% else

        dim l_marker_vals
        l_marker_vals = "Y,N"

        %>
        <tr valign="top">
          <td align="left" width="640" colspan="3"></td>
        </tr>
        <tr valign="top">
          <td align="left" width="640" colspan="3">
            <p><noBr>
              
            <input type="image" src="..\graphics\submit1222.gif" width="54" height="16" border="0" alt="Submit to Database" id=image1
            name=image1>
            <font face="Verdana, Arial, Helvetica, sans-serif" size="4">
              <a href="admin6a.asp">
                Return to Panel</a></font></noBr> </p>
          </td>
        </tr>
      </table>
    </body>
  </html>
```

```

<table border="0" cellspacing="3" cellpadding="2">
<tr>
<td bgcolor="003366" height="25" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">
PANEL NAME
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
PANEL START DATE
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
PANEL TYPE
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
PANEL SUBTYPE
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
PANEL TEST TYPE
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
PACKAGE
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
EXPENSE SET
</font></b></font></div></td>

<td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
REVENUE SET
</font></b></font></div></td>
<%><td bgcolor="003366" height="25" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
NEWSSTAND MARKER
</font></b></font></div></td>
</tr>

<%>

<% for i = 0 to l_size - 1 %>
<tr>

```



```
<td height="15" align="left">
  <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
    <input type="text" name="P_PANEL_<%Response.write i %>" maxlength="250" size="30">
  </font></b></font></div>
</td>
<td height="15" align="left">
  <div align="left">
    <input type="text" name="P_PANEL_DATE_START<% Response.Write i %>" size="12" maxlength="12">
  </div>
</td>

<td height="15" align="left">
  <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
    <b><font size="1" color="#000000">

    <% 'strSQL = "select PANEL_TYPE_NAME from CORP_PANEL_TYPE"
      'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
      showLOVListBox PanelTypeLOV, "P_PANEL_TYPE" & i,4, "SELECT TYPE", -1
      'objRST.close

      %>

      </font></b></font></div>
    </td>

    <td height="15" align="left">
      <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
        <% 'strSQL = "select PANEL_SUB_TYPE_NAME from CORP_PANEL_SUB_TYPE"
          'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
          showLOVListBox PanelSubTypeLOV, "P_PANEL_SUB_TYPE" & i,4, "SELECT
SUBTYPE", -1
          showQueryListBox objRST, "P_PANEL_SUB_TYPE" & i,4, "SELECT SUBTYPE", -
1
          'objRST.close

          %>

          </font></b></font></div>
        </td>

        <td height="15" align="left">
          <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
            <% 'strSQL = "select PANEL_TEST_TYPE_NAME from CORP_PANEL_TEST_TYPE"
              'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
              showLOVListBox PanelTestTypeLOV, "P_TEST_TYPE" & i,4, "SELECT TEST
TYPE", -1
              'objRST.close

              %>

              </font></b></font></div>
            </td>

            <td height="15" align="left">
              <div align="left">
                <% 'strSQL = "select PACKAGE_NAME from PROD_PANEL_PACKAGE WHERE Product_id =" &
Session("PRODUCT_ID")
                'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
                showLOVListBox PackageLOV, "P_PANEL_PKG" & i,4, "SELECT PACKAGE", -1
                'objRST.close

                %>

                </div>
              </td>

              <td height="15" align="left">
```





```
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
    <% strSQL = "select e_set_name from EXPENSE_SETS where "
        & " PRODUCT_ID=" & Session("PRODUCT_ID")
        'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        showLOVListBox expenseSetLOV, "P_E_SET_NAME" & i,6, "SELECT EXPENSE
SET",l_e_Set_name
        'objRST.close %>
        </font></b></font></div>

</td>

<td height="15" align="left">
    <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
    <% strSQL = "select r_set_name from REVENUE_SETS where "
        & " PRODUCT_ID=" & Session("PRODUCT_ID")
        'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        showLOVListBox RevenueSetLOV, "p_r_SET_NAME" & i,6, "SELECT REVENUE
SET",l_r_set_name
        'objRST.close %>

    </font></b></font></div>
    </td>
    <% ' <TD align="left">
        'showLOVListBox l_marker_vals, "P_NEWSTAND_SALE_MARKER" & i, 6, "----",l_newstand_sale_marker
        '
    </td>
    <%>
</tr>

<% next %>

</table>
<p>&nbsp;</p>
<% end if %>
</td>
</tr>
</table>
</form>
</body>
</html>
```



addpanelpackage.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addPanelPackage.asp"
l_col_names(0) = "PANEL PACKAGE"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Prod_panel_Package"
l_Title = " Panel Package"
l_setupTabLink = "setupPanelPackage.asp"
l_showPrevSelection= false
l_file = "addPanelPackage.asp"

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_ColumnCount
addBottom
%>
```

addpanelsubtype.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addPanelSubType.asp"
l_col_names(0) = "PANEL SUB TYPE"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Corp_Panel_Sub_Type"
l_Title = " Panel Sub Type "
l_setupTabLink = "setupPanelSubType.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, -1
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_ColumnCount
addBottom %>
```



addpaneltesttype.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addPanelTestType.asp"
l_col_names(0) = "PANEL TEST TYPE"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Corp_Panel_Test_Type"
l_Title = " Panel Test Type "
l_setupTabLink = "setupPanelTestType.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, -1
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names , l_ColumnCount
addBottom %>
```



addpaneltype.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addPanelType.asp"
l_col_names(0) = "PANEL TYPE"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Corp_Panel_Type"
l_Title = " Panel Type "
l_setupTabLink = "setupPanelType.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, -1
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names , l_ColumnCount
addBottom %>
```

addpriorsrcdesc.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<!-- #include file="../support/libCommon.asp" -->

<%

dim l_col_names(8)
l_file = "addPriorSrcDesc.asp"
l_col_names(0) = "PRIOR SOURCE IDENTIFIER"
l_col_names(1) = "RENEWALS PRIOR SOURCE DESC"
l_col_names(2) = "BILLING CURRENT SOURCE DESC"
l_col_names(3) = "MODEL SOURCE CATEGORY"
l_col_names(4) = "MODEL SOURCE"
l_col_names(5) = "AGENT SOURCE"
l_col_names(6) = "VALID FROM DATE"
l_col_names(7) = "VALID TO DATE"

l_ColumnCount = 8

l_sp_add = "sp_Add_prior_Source_desc"
l_Title = "Prior Source Description"
l_setupTabLink = "setupPriorSrcDesc.asp"
l_showPrevSelection= true

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, Session("PRODUCT_ID")
addHeading l_file, l_file
'addMain l_title, l_setupTabLink, l_columncount, "", l_col_names, control_type, control_arg1, control_arg2,
control_arg3, showPrevSelection
addMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_ColumnCount
addBottom

%>
```

addrset.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/pickListObj.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->

<%
'Response.Write "Action " & "<br>"
dim DEBUG
DEBUG = false

dim l_label(10)
dim l_values(33)
'dim l_basis_opt(10)(7)
    l_label(0) = "Sub Revenue"
    l_label(1) = "Ad Revenue"
    l_label(2) = "List Rental Revenue"
    l_label(3) = "Other Revenue"

If Request.Form("SUBMIT.x") <> "" then
    l_action = "SUBMIT"
else
    l_action = "x"
end if
'Response.Write "Action " & l_action & "<br>"
if l_action = "SUBMIT" then
    'response.write "<BR>Add invoked<BR>"
    for i = 0 to 7
        l_values(i) = Request.Form("P_" & i)
    next
    l_r_set_name = Request.Form("P_r_SET_NAME")
    if l_r_set_name > "" then
        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_add_prod_r_set"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        '        USER SUPPLIED
        objCMD.Parameters(1).Value = Session("PRODUCT_ID")
        objCMD.Parameters(2).Value = l_r_set_name
        for k = 0 to 7
            IF DEBUG = TRUE THEN
                Response.Write (k + 3) & " " & k & " " & l_values(k) & "<br>"
            END IF
            if k mod 2 = 0 then
                objCMD.Parameters(k + 3).Value = validateNumber(l_values(k))
            else
                objCMD.Parameters(k + 3).Value = validatePickList(l_values(k))
            end if
        ' Response.Write k + 3 & " " & l_values(k) & "<br>"
        next
        Set TempNonSessionVar = objCMD.Execute()
    end if
    Response.Redirect "setupRset.asp"
Else
    l_action = "New"
end if
```

```

'Response.Write "Hate " & l_action
%>
<HTML><HEAD><title>
addRset.asp
</title>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<script language="JavaScript">
<!--
function MM_swapImgRestore() { //v2.0
  if (document.MM_swapImgData != null)
    for (var i=0; i<(document.MM_swapImgData.length-1); i+=2)
      document.MM_swapImgData[i].src = document.MM_swapImgData[i+1];
}

function MM_preloadImages() { //v2.0
  if (document.images) {
    var imgFiles = MM_preloadImages.arguments;
    if (document.preloadArray==null) document.preloadArray = new Array();
    var i = document.preloadArray.length;
    with (document) for (var j=0; j<imgFiles.length; j++) if (imgFiles[j].charAt(0)!="#"){
      preloadArray[i] = new Image;
      preloadArray[i++].src = imgFiles[j];
    }
  }
}

function MM_swapImage() { //v2.0
  var i,j=0,objStr,obj,swapArray=new Array,oldArray=document.MM_swapImgData;
  for (i=0; i < (MM_swapImage.arguments.length-2); i+=3) {
    objStr = MM_swapImage.arguments[(navigator.appName == 'Netscape')?i:i+1];
    if ((objStr.indexOf('document.layers[')==0 && document.layers==null) ||
        (objStr.indexOf('document.all[') ==0 && document.all ==null))
      objStr = 'document'+objStr.substring(objStr.lastIndexOf('.'),objStr.length);
    obj = eval(objStr);
    if (obj != null) {
      swapArray[j++] = obj;
      swapArray[j++] = (oldArray==null || oldArray[j-1]!=obj)?obj.src:oldArray[j];
      obj.src = MM_swapImage.arguments[i+2];
    }
  }
  document.MM_swapImgData = swapArray; //used for restore
}
//-->
</script>
<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="addRSet.asp" id=form1 name=form1>
<table width="320" border="0">
<tr>
<td colspan="3">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="25%" valign="top"></td>
<td width="75%" colspan="3" valign="top">
<div align="right"></div>
<div align="right"></div>
</td>
</tr>
<tr>
<td width="25%" valign="top">&nbsp;</td>
<td width="75%" colspan="3" valign="top">
<div align="right"></div>
<div align="right">
<p>
<input type="image" name="SUBMIT" value="SUBMIT" src="../graphics/submit1222.gif" width="54" height="16"
border="0"

```



```

alt="Submit to Database" id=image1>
  </p>
  <p> <font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="setupRset.asp">Return
    to Revenue Set Setup Table</a></font> </p>
  </div>
  </td>
</tr>
</table>
</td>
</tr>
<tr>
  <td height="10" align="center" width="150">&nbsp;</td>
  <td height="10" colspan="2">&nbsp;</td>
</tr>
<tr>
  <td height="20" bgcolor="003366" align="center" width="150"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1">Revenue
  Set </font></b></font></td>
  <td height="20" colspan="2">
    <input type="TEXT" name='p_r_set_name'
      id=0 value=" " size="27">
  </td>
</tr>
<% for i=0 to 3 %>
  <tr>
    <td height="20" bgcolor="003366" align="center" width="150"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1">
      <%
        Response.Write l_label(i)
      %>
    </font></b></font></td>
    <%
      for j = 0 to 1
        in_cnt = i * 2 + j
      %>
    <td height="20">
      <% if j = 1 then %>
      <select name="P_<%Response.write in_cnt %>">
        <% if l_values(in_cnt) = "CPM" then %>
          <OPTION value=0>CPM or CPU
          <OPTION value =CPM'selected>CPM</option>
          <OPTION value =CPU' >CPU</option>
        <% elseif l_values(in_cnt) = "CPU" then %>
          <OPTION value=0>CPM or CPU
          <OPTION value =CPM'>CPM</option>
          <OPTION value =CPU' selected>CPU</option>
        <% else %>
          <OPTION value=0>CPM or CPU
          <OPTION value =CPM'>CPM</option>
          <OPTION value =CPU'>CPU</option>
        <% END IF %>
      </select>
      <% elseif j = 2 then %>
      <input type="TEXT" name="P_<%Response.write in_cnt %>" size="12" maxlength="12">
    </td>
    <% end if %>
    <% next %>
  </tr>
<% next %>

```



```
</table>
<p>&nbsp;</p>
</form>
</BODY>
</HTML>
```

addsource.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%

on error resume next
dim p_source(4)
dim p_src_ind(4)
added = false
errors = false
for i = 0 to 4

p_source(i) = REQUEST("p_source" & cstr(i) )
p_src_ind(i) = REQUEST("p_src_ind" & cstr(i) )
' Response.Write p_source(i) & p_src_ind(i)
if p_source(i) > " " and p_src_ind(i) > " " then
    added = true
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_add_source"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    objCMD.Parameters(1).Value = p_source(i)
    objCMD.Parameters(2).Value = p_src_ind(i)
    objCMD.Parameters(3).Value = cint(Session("PRODUCT_ID"))
    objCMD.Parameters(4).Value = Session("SRC_CAT_NAME")
    objCMD.Parameters(5).Value = Session("GRP_NAME")
    objCMD.Parameters(6).Value = Session("TITLE_NAME")
    Set TempNonSessionVar = objCMD.Execute()
end if
next

if added = true then
    Response.Redirect ("admin4a.asp" )
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="addSource.asp" name=form1 id=form1>
<table width="640" border="0" cellpadding="2">

<% prevSelections (3) %>

<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<p><nobr>
</p>
</td>
</tr>
</table>
</form>
</body>
</html>
```



```
<input type="image" src="../graphics/submit1222.gif" width="54" height="16" border="0" alt="Submit to Database">
<font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="admin4a.asp">Return
to Source</a></font></nobr> </p>
<table width="265" border="0" cellspacing="3" cellpadding="2">
<tr>
<td bgcolor="003366" height="25" width="166" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">SOURCE
NAME </font></b></font></div>
</td>
<td bgcolor="003366" height="25" width="41" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font
size="1">SOURCE<br>
INDICATOR </font></b></font></div>
</td>
</tr>

<% for i = 0 to 4 %>
<tr>
<td width="166" height="15" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">
<%
strSQL = "select U_SOURCE_NAME FROM Source_descriptions "
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SOURCE" & i,4, "SELECT SOURCE", -1
objRST.close

%>

</font></b></font></div>
</td>
<td width="41" height="15" align="left">
<div align="left">
<input type="text" name="p_src_ind<% Response.Write i %>" size="10" maxlength="10">
</div>
</td>
</tr>
<% next %>

</table>
<p>&nbsp;</p>

</td>
</tr>
</table>
</form>
</body>
</html>
```



addsourcename.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%

dim l_col_names(2)
l_file = "addSourceName.asp"
l_col_names(0) = "SOURCE"
l_col_names(1) = "DESCRIPTION"
l_ColumnCount = 2
l_sp_add = "sp_Add_Source_Description"
l_Title = "Source Description "
l_setupTabLink = "setupSourceName.asp"
l_showPrevSelection= false

addAccept l_ColumnCount, l_setupTabLink, l_sp_add, -l
addHeading l_file, l_file
addMain l_Title, l_setupTabLink, l_showPrevSelection, L_Col_names , l_ColumnCount
addBottom %>
```

admin1a.asp

```
<%@ Language=VBScript %>
```

```
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<%
```

```
Msg = Request.QueryString("MSG")
```

```
if not session("PID_PHRASE") > " " then
    Response.Redirect("/welcome.asp")
end if
```

```
l_this_file = "admin1a.asp"
if (Request.Form("display") = "2"
    and Request.Form("setupTab") <> "0"
    and Request.Form("setupTab") > " ") then
    Response.Redirect(Request.Form("setupTab"))
end if
```

```
dim tmp
tmp = Request.Form("cmb")
Request.Form("cmb")
IF tmp <> "0" and tmp > " " THEN
    session("GRP_NAME") = tmp
    Response.Redirect("admin2a.asp")
end if
%>
```

```
<html>
<head>
<title>Admin1a.asp</title>
```

```
</head>
```

```
<body bgcolor="#99CCFF" vlink="#000000" link="#003366">
    <form method="post" action="<%Response.Write l_this_file %>" id="form2" name="form2">
```

```
    <table width="640" border="0" cellpadding="2">
        <tr valign="top">
```

```
            <td width="65">
                <% ' <a href="adminSearch.asp">
                '  </a> %>
                </td>
                <td width="425">
                    <p> <font color="black" size="-1" face="Verdana, Arial, Helvetica, sans-serif" color=black>
                    <% if Msg = "YES" then %>
                    You have been redirected to this page because <br>
                    the session variables needed to perform the action<br> you requested have expired.<br>
                    Please select the values again.<br>
                    </font>
                    <% end if %>
```

```
                <br>
```

```
            <%
```

```
strSQL = "select distinct GRP_NAME from CORP_INFO WHERE " & SESSION("PID_PHRASE")
' Response.Write strSQL & "<BR>"
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
```



```
showQueryListBox objRST, "cmb",5, "SELECT PRODUCT GROUP", " "
objRST.close
%>
<input type="IMAGE" name="go" value="go" src="../graphics/go1222.gif" WIDTH="33" HEIGHT="16" border=0>

</p>

</td>

</form>
</tr>
<tr valign="top">
<td width="640" colspan="3">
<p>&nbsp;</p>

</td>
</tr>
</table>
</form>
</body>
</html>
```

admin2a.asp

```
<%@ Language=VBScript %>

<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoutils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<%
Session("USR_ACCESS") = "NONE"
l_from = Request("FR")
if not (l_from > " ") then
    checkSessionVars(1)
end if

l_this_file = "admin2a.asp"

if (Request.Form("display")="2"
    and Request.Form("setupTab") <> "0"
    and Request.Form("setupTab") > " ") then
    Response.Redirect(Request.Form("setupTab"))
end if

dim tmp
tmp = Request.Form("cmb")
'response.write ("<br>Product: " & tmp & "<br>")

'response.write tmp & " this don't work"
IF tmp <> "0" and tmp > " " and Request.Form("from") = "0" THEN
    session("PRODUCT_ID") = tmp
    strSQL = "select TITLE_NAME from CORP_INFO where grp_name=" & SESSION("GRP_NAME") _
        & " and Product_id=" & tmp
    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    session("TITLE_NAME") = objRST.Fields(0).Value
    objRST.close
    'session("TITLE_NAME") =
        check_security
    Response.Redirect("admin3a.asp")
else
    if Request.Form("from") > " " then
        l_from = Request.Form("from")
    end if
    IF tmp <> "0" and tmp > " " then

        session("PRODUCT_ID") = tmp
        strSQL = "select TITLE_NAME, GRP_NAME from CORP_INFO where" _
            & " Product_id=" & tmp
        objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

        session("TITLE_NAME") = objRST.Fields(0).Value
        session("GRP_NAME") = objRST.Fields(1).Value

        Response.Redirect(l_from)
        objRST.close
        end if
    end if
    if not l_from > " " then
        Session("refer_page") = "admin2a.asp"
        Session("refer_Title") = "Product"
    end if
%>

<html>
<head>
<title>Admin2a.asp</title>
```



```

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
  <form method="post" action="admin2a.asp" id="form2" name="form2">
    <table width="640" border="0" cellpadding="2">
      <% if l_from > " " then %>
        <font size="4" face="Verdana, Arial, Helvetica, sans-serif" color=black>
          <b>Please select a Product Id </b>
        </font>
        <input type="Hidden" name="FR" value="<%Response.Write l_from %>">
      <% else %>
        <% prevSelections (1) %>
        <input type="Hidden" name="from" value="0">
      <% end if %>
    <tr valign="BOTTOM">
      <td width="65">
      </td>
      <td width="425">
        <p> <br>
        <% strGrp = session("GRP_NAME")
          if l_from > " " then
            strSQL = "select PRODUCT_ID, TITLE_NAME from
CORP_INFO where " _
                        & " " & SESSION("PID_PHRASE") _
                        & " Order by title_name "
          else
            strSQL = "select PRODUCT_ID, TITLE_NAME from
CORP_INFO where " _
                        & " grp_name=" & strGrp _
                        & " and ( " & SESSION("PID_PHRASE") & ")" _
                        & " Order by title_name "
          end if
          'response.write ("<br>SQL: " & strSQL & "<br>")
          objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
          showQueryListBox objRST, "cmb",5, "SELECT PRODUCT", "
          objRST.close %>
          <input type="IMAGE" name="go" value="go" src=" ../graphics/go1222.gif"
border=0>
        </p>
      </td>
    </tr>
    <tr>
      <td width="*">
        <%> 
        ' setupTableListBox %>
        <br>
      </td>
    </tr>
    <tr valign="top">
      <td width="640" colspan="3">
        <p>&nbsp;</p>
      </td>
    </tr>
  </table>
</form>
</body>
</html>

```

admin3a.asp

```
<%@ Language=VBScript %>

<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->

<%
'checkSessionVars(2)
l_this_file = "admin3a.asp"
Session("refer_page") = "admin3a.asp"
Session("refer_Title") = "Source Category"

'if Request.Form("AUTOSSETUP") <> "" then response.redirect("tabautosetup.asp")
'if (Request.Form("display")="2"
' and Request.Form("setupTab")<> "0"
' and Request.Form("setupTab") > " ") then
' Response.Redirect(Request.Form("setupTab"))
'end if
dim tmp
tmp = Request.Form("cmb")
'response.write ("<br>Source Category: " & tmp & "<br>")
IF tmp <> "0" and tmp > " " THEN
    Session("SRC_CAT_NAME") = tmp
    Response.Redirect("admin4a.asp")
end if
%>

<html>
<head>
<title>Admin3a.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<table width="640" border="0" cellpadding="2">
<% prevSelections (2) %>
<form method="post" action="<%Response.Write l_this_file %>" id="form2" name="form2">
<tr valign="BOTTOM">
<td width="65">
<% ' <a href="adminSearch.asp">
' </a>
%>
</td>

<td width="425">
<br>

<%

strSQL = "select SRC_CAT_NAME from SRC_CATEGORY where " _
& " grp_name = " & Session("GRP_NAME") & ""
& " and PRODUCT_ID=" & Session("PRODUCT_ID") & ""

objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

showQueryListBox objRST, "cmb",5, "SELECT SOURCE CATEGORY",-1
objRST.close
%>
<input type="IMAGE" name="go" value="go" src="..\graphics/go1222.gif" WIDTH="33" HEIGHT="16" border="0">
<p>
```



```
<p>
<input type="HIDDEN" name="AUTOSETUP" value="Edit Auto-Setup Parameters">
<p>
```

```
</td>
</form>
```

```
<% setupTableListBox l_this_file%>
```

```
</tr>
<tr valign="top">
  <td width="640" colspan="3">
    <p>&nbsp;</p>
  </td>
</tr>
</table>
</form>
</body>
</html>
```

admin4a.asp

```
<%@ Language=VBScript %>

<!-- #include file="./support/global_constants.asp" -->
<!-- #include file="./support/dbConnection.asp" -->
<!-- #include file="./support/adoUtils.asp" -->
<!-- #include file="./support/srcTab.asp" -->
<!-- #include file="./support/JavaScripts.asp" -->

<%
checkSessionVars(3)
l_this_file = "admin4a.asp"

Session("refer_page") = "admin4a.asp"
Session("refer_Title") = "Source"
if (Request.Form("display")="2" _
    and Request.Form("setupTab") <> "0" _
    and Request.Form("setupTab") > " ") then
    Response.Redirect(Request.Form("setupTab"))
end if

dim tmp
tmp = Request.Form("cmb")

IF tmp <> "0" and tmp > " " THEN
    session("SOURCE_NAME") = tmp
    strSQL = "select SRC_INDICATOR from SOURCE where SOURCE_NAME=" & SESSION("SOURCE_NAME") _
        & " and Product_id=" & SESSION("PRODUCT_ID")
    'Response.Write strSQL
    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    session("SRC_INDICATOR") = objRST.Fields(0).Value
    objRST.close
    'session("TITLE_NAME") =
    Response.Redirect("admin5a.asp")
end if

%>
<head><title>Admin4a.asp</title>
<% LorieScripts %>
</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366"
onLoad="MM_preloadImages(graphics/edittablero1222.gif,#945899269583');MM_preloadImages(graphics/edittablero1222.gif,#945899269583');">
<% ' <body bgcolor="#99CCFF" vlink="000000" link="003366"> %>

<table width="640" border="0" cellpadding="2">
<% prevSelections (3) %>

    <form method="post" action="<%Response.Write l_this_file %>" id="form2" name="form2">
<tr valign="BOTTOM">
<td width="65">
    <% ' <a href="adminSearch.asp">
    ' </a>
    %>
</td>
<td width="500">
    <p> <br>

    <%

strSQL = "select SOURCE_NAME from SOURCE where " _
    & " grp_name = " & Session("GRP_NAME") & " " _
```



```
& " and PRODUCT_ID=" & Session("PRODUCT_ID") _
& " and src_cat_name = " & Session("SRC_CAT_NAME") & ""

objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

showQueryListBox objRST, "cmb", 5, "SELECT SOURCE", -1
objRST.close
%>

```

admin5a.asp

```
<%@ Language=VBScript %>

<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->
<!-- #include file="../support/objEdit.asp" -->
<%

checkSessionVars(4)
l_this_file = "admin5a.asp"

Session("refer_page") = "admin5a.asp"
Session("refer_Title") = "Campaign"

if (Request.Form("display")="2"
    and Request.Form("setupTab") <> "0"
    and Request.Form("setupTab") > " ") then
    Response.Redirect(Request.Form("setupTab"))
end if

dim tmp

tmp = Request.Form("cmb")
IF tmp <> "0" and tmp > " " THEN
    session("CAMP_NAME") = tmp
    Response.Redirect("admin6a.asp")
end if

'Dim vars

'Common
dim recordcount
dim checked_records
dim i
dim WhereClause
dim recordidlist

'Edit vars
dim MSE

'Group Edit vars
dim PGE
dim index
dim recordid

'Add
if Request.form("addbutton") > " " then response.redirect("addCamp.asp")

'Edit
if Request.form("editbutton") > " " then
    'Setup first page, then let edit handle the rest

    'init obj
    set MSE = new objGroupEdit

    'Get records
    recordcount = Request.form("P_RecCount")
    'response.write("There are " & cstr(recordcount) & " records <br>")
    recordidlist = ""
    toomanyrecords = 0
    checked_records = 0
    for i = 1 to recordcount
        recordid = Request.form("P_check" & cstr(i))
        if recordid > " " and len(recordidlist) > 0 then
```

```

        checked_records = checked_records + 1
        if checked_records > 19 then
            toomanyrecords = 1
        else
            recordidlist = recordidlist + "," + cstr(recordid)
        end if
    elseif recordid > " " then 'don't add comma first time
        recordidlist = cstr(recordid)
    end if
next
Session("RECORDIDLIST") = recordidlist
Session("toomanyrecords") = toomanyrecords
if recordidlist > " " then response.redirect("ms_editcampaign.asp?pg=1")
end if

'Post/Delete/Copy Group Edit Info
if Request.form("GroupEdit") > " " or request.form("deletebutton") > " " or request.form("copybutton") > " " then

    'init obj
    set PGE = new objPostGroupEdit
    PGE.init

    'Get records
    recordcount = Request.form("p_RecCount")

    'response.write("There are " & cstr(recordcount) & " records <br>")
    for i = 1 to recordcount
        recordid = Request.form("P_check" & cstr(i))
        if recordid > " " then PGE.AddRecord recordid
        response.write(recordid)
    next i

    'Set Stored Procedure info
    PGE.ClassType = "Group Edit"
    PGE.SP = "sp_edit_campaign"
    PGE.TableName = "campaigns"
    if Request.form("GroupEdit") > " " then PGE.IsDelete = 0
    if request.form("deletebutton") > " " then PGE.IsDelete = 1
    if request.form("copybutton") > " " then PGE.IsDelete = 2
    PGE.AddParam "camp_name"
    PGE.AddParam "camp_date_fr"
    PGE.AddParam "camp_type"
    PGE.AddParam "camp_status_name"
    PGE.AddParam "e_set_name"
    PGE.AddParam "r_set_name"
    'PGE.AddParam "newstand_sale"
    PGE.AddParam "budgeted_gross_pct"
    PGE.AddParam "budgeted_vol"
    PGE.AddParam "login"
    PGE.AddParam "rec_id"
    PGE.AddParam "IsDelete"
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_edit_campaign"
    objCMD.CommandType = adCmdStoredProc
    objCMD.parameters.refresh
    PGE.PostGroupEditsBySP
end if
%>

<html>
<head>
<title>Admin5a.asp</title>

</head>
<% LorieScripts %>
<body bgcolor="#99CCFF" vlink="000000" link="003366">

<table width="640" border="0" cellpadding="2">

```



```
<% prevSelections (4) %>
<form method="post" action="<%Response.Write l_this_file %>" id="form2" name="form2">
<tr valign="BOTTOM">
<td width="65">
<% ' <a href="adminSearch.asp">
' </a>
%>
</td>
<td width="425">
<p> <br>

<%

    strSQL = "select CAMP_NAME from CAMPAIGNS where " _
    & " grp_name = " & Session("GRP_NAME") & "" _
    & " and PRODUCT_ID=" & Session("PRODUCT_ID") & "" _
    & " and src_cat_name = " & Session("SRC_CAT_NAME") & "" _
    & " and source_name = " & Session("SOURCE_NAME") & "" _
    & " ORder by CAMP_NAME DESC "

objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

showQueryListBox objRST, "cmb",5, "SELECT CAMPAIGN", -1
objRST.close

%> <input type="IMAGE" name="go" value="go" src=" ../graphics/go1222.gif"
WIDTH="33" HEIGHT="16" border=0>

</p>

</td>
</form>

<% ' setupTableListBox l_this_file %>

</tr>
<tr valign="top">
<td width="640" colspan="3">
<p>&nbsp;</p>

</td>
</tr>
</table>
<%
'Group Edit info
    set GE = new objGroupEdit
    GE.init
    GE.action = "admin5a.asp"
    GE.RowCount = 1

'Group Edit Show/Hide button
if Request.form("SHOWGROUPEDIT") > " " then
    %> <form method="post" action=" <%Response.Write l_this_file %>" id="hidegeform"
name="hidegeform">
        <table>
            <tr>
                <td width="20">&nbsp;</td>
                <td><input type="submit" value="Hide Group Edit"
name="HIDEGROUPEDIT" action="form.submit"></td>
```





```

                                </tr>
                                </table>
                                </form>
                                <%
                                GE.Show = True
                                else
                                %> <form method="post" action="<%Response.Write l_this_file %>" id="showgeform"
name="showgeform">
                                <table>
                                <tr>
                                <td width="20">&nbsp;</td>
                                <td><input type="submit" value="Show Group Edit"
name="SHOWGROUPEdit" action="form.submit"></td>
                                </tr>
                                </table>
                                </form>
                                <br> <form method="POST" action="<%=action%>" name="Group_Edit_Form" cellpadding="3"
cellspacing="3">
                                <%
                                GE.Show = False
                                end if

                                'Campaign Name
                                index = GE.AddItem
                                GE.fieldType index, "LABEL"
                                GE.ColumnHeader index, "CAMPAIGN NAME"

                                'Campaign Start Date
                                index = GE.AddItem
                                GE.fieldType index, "TEXT"
                                GE.ColumnHeader index, "CAMPAIGN DATE"

                                'Campaign Type
                                index = GE.AddItem
                                GE.fieldType index, "SELECT"
                                GE.ColumnHeader index, "CAMPAIGN TYPE"
                                GE strSQL index, "select CAMP_TYPE_NAME from PROD_CAMP_TYPE where PRODUCT_ID=" &
Session("PRODUCT_ID") & " order by camp_type_name "

                                'Campaign Status
                                index = GE.AddItem
                                GE.fieldType index, "SELECT"
                                GE.ColumnHeader index, "CAMPAIGN STATUS"
                                GE strSQL index, "select CAMP_STATUS_NAME from PROD_CAMP_STATUS where PRODUCT_ID=" &
Session("PRODUCT_ID") & " order by camp_status_name "

                                'Campaign End Date
                                index = GE.AddItem
                                GE.fieldType index, "LABEL"
                                GE.ColumnHeader index, "CAMPAIGN END DATE"

                                'Expense Set
                                index = GE.AddItem
                                GE.fieldType index, "SELECT"
                                GE.ColumnHeader index, "EXPENSE SET"
                                GE strSQL index, "select e_set_name from EXPENSE_SETS where PRODUCT_ID=" &
Session("PRODUCT_ID") & " order by e_set_name "

                                'Revenue Set
                                index = GE.AddItem
                                GE.fieldType index, "SELECT"
                                GE.ColumnHeader index, "REVENUE SET"
                                GE strSQL index, "SELECT r_set_name from revenue_sets where product_id = " & Session("PRODUCT_ID")
                                & " order by r_set_name"

                                'Newstand Sale
                                index = GE.AddItem
                                GE.fieldType index, "TEXT"
```

```

'GE.ColumnHeader index, "NEWSTAND SALE"

'Budgeted Gross %
index = GE.AddItem
GE.fieldtype index, "TEXT"
GE.ColumnHeader index, "BUDGETED GROSS PCT"

'Budgeted Vol
index = GE.AddItem
GE.fieldtype index, "TEXT"
GE.ColumnHeader index, "BUDGETED VOL"

'Page
GE.MS_Page = 1

'Display Table
GE.DisplayEditTable

l_query = " select CAMP_NAME as 'CAMPAIGN NAME', CAMP_DATE_FR as 'CAMPAIGN DATE', CAMP_TYPE
as 'CAMPAIGN TYPE', " _
& " CAMP_STATUS_NAME as 'CAMPAIGN STATUS', E_SET_NAME as 'EXPENSE SET', R_SET_NAME as 'REVENUE
SET', " _
& " BUDGETED_GROSS_PCT as 'BUDGETED GROSS PCT', BUDGETED_VOL as 'BUDGETED VOLUME',
REC_ID from CAMPAIGNS where " _
& " GRP_NAME = " & Session("GRP_NAME") & " " _
& " and PRODUCT_ID = " & Session("PRODUCT_ID") & " " _
& " and src_cat_name = " & Session("SRC_CAT_NAME") & " " _
& " and source_name = " & Session("SOURCE_NAME") & " " _
& " Order by CAMP_NAME DESC "

l_columnCount = 8
l_addProc = "addCamp.asp"
l_editProc = "editCamp.asp"
dim l_title(8), l_columnWidth(8)

l_title(0) = "CAMPAIGN NAME "
l_columnWidth(0) = "152"
l_title(1) = "CAMPAIGN<br> START DATE "
l_columnWidth(1) = "76"
l_title(2) = "CAMPAIGN TYPE "
l_columnWidth(2) = "76"
l_title(3) = "STATUS "
l_columnWidth(3) = "69"
l_title(4) = "EXPENSE SET "
l_columnWidth(4) = "73"
l_title(5) = "REVENUE SET "
l_columnWidth(5) = "73"
l_title(6) = "BUDGETED GROSS PCT"
l_columnWidth(6) = "40"
l_title(7) = "BUDGETED VOLUME"
l_columnWidth(7) = "40"

'adminTable l_title, l_query, l_editProc, l_addProc, l_columnWidth, l_columnCount
l_name = "Campaigns"
l_editProc = "editCamp.asp"

l_addProc = "addCamp.asp"
set myObjEdit = New ObjEdit
myObjEdit.init l_name, l_editProc, l_addProc, l_this_file
myObjEdit.defineComplex(8)
myObjEdit.setQuery(l_query)
myObjEdit.SelectTable

%>
</form>
</body>
</html>

```

admin6a.asp

```
<%@ Language=VBScript %>

<!-- #include file=../support/global_constants.asp -->
<!-- #include file=../support/dbConnection.asp -->
<!-- #include file=../support/adoUtils.asp -->
<!-- #include file=../support/srcTab.asp -->
<!-- #include file=../support/JavaScripts.asp -->
<!-- #include file=../support/objEdit.asp -->
<!-- #include file=../support/objRecIdManager.asp -->
<%
checkSessionVars(5)
l_this_file = "admin6a.asp"

Session("refer_page") = "admin6a.asp"
Session("refer_Title") = "Panel"

if (Request.Form("display")="2" and Request.Form("setupTab")<> "0" and Request.Form("setupTab") > " ") then
Response.Redirect(Request.Form("setupTab"))
dim tmp
tmp = Request.Form("cmb")

IF tmp <> "0" and tmp > " " THEN
    session("PANEL_NAME") = tmp
    Response.Redirect("AdminKeys.htm")
end if

'Dim vars

'Common
dim recordcount
dim i
dim WhereClause
dim recordidlist

'Edit vars
dim MSE

'Group Edit vars
dim PGE
dim index
dim recordid

'Add
if Request.form("addbutton") > " " then response.redirect("addPanel.asp")

'Edit
if Request.form("editbutton") > " " then
    'Setup first page, then let edit handle the rest

    'init obj
    set MSE = new objGroupEdit

    'Get records
    recordcount = Request.form("P_RecCount")

    'response.write("There are " & cstr(recordcount) & " records <br>")
    recordidlist = ""
    toomanyrecords = 0
    checked_records = 0
    for i = 1 to recordcount
        recordid = Request.form("P_check" & cstr(i))
        if recordid > " " and len(recordidlist) > 0 then
            checked_records = checked_records + 1
            if checked_records > 19 then
                toomanyrecords = 1
            else

```

```

                                recordidlist = recordidlist + "," + cstr(recordid)
                                end if
                                elseif recordid > " " then ' don't add comma first time
                                    recordidlist = cstr(recordid)
                                end if
                            next
                            Session("RECORDIDLIST") = recordidlist
                            Session("toomanyrecords") = toomanyrecords
                            session("page") = 1
                            if recordidlist > " " then response.redirect("ms_editpanel.asp?pg=1")
                        end if

'Post/Delete Group Edit Info
if Request.form("GroupEdit") > " " or request.form("deletebutton") > " " or request.form("copybutton") > " " then

    'init obj
    set PGE = new objPostGroupEdit
    PGE.init

    'Get records
    recordcount = Request.form("p_RecCount")

    'response.write("There are " & cstr(recordcount) & " records <br>")
    for i = 1 to recordcount
        recordid = Request.form("P_check" & cstr(i))
        if recordid > " " then PGE.AddRecord recordid
        response.write(recordid)
    next 'i

    'Set Stored Procedure info
    PGE.ClassType = "Group Edit"
    PGE.SP = "sp_edit_panel"
    PGE.TableName = "panels"
    if Request.form("GroupEdit") > " " then PGE.IsDelete = 0
    if request.form("deletebutton") > " " then PGE.IsDelete = 1
    if request.form("copybutton") > " " then PGE.IsDelete = 2
    PGE.AddParameter "panel_name"
    PGE.AddParameter "panel_date_start"
    PGE.AddParameter "panel_type"
    PGE.AddParameter "panel_subtype"
    PGE.AddParameter "test_type"
    PGE.AddParameter "panel_pkg"
    PGE.AddParameter "e_set_name"
    PGE.AddParameter "r_set_name"
    PGE.AddParameter "newstand_sale_marker"
    PGE.AddParameter "login"
    PGE.AddParameter "rec_id"
    PGE.AddParameter "IsDelete"
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_edit_panel"
    objCMD.CommandType = adCmdStoredProc
    objCMD.parameters.refresh
    PGE.PostGroupEditsBySP

'Post Edits By Query starts here ...
    'Build where clause
    ' Whereclause = " and grp_name = " & Session("GRP_NAME") & ""
    ' & " and PRODUCT_ID=" & Session("PRODUCT_ID") & ""
    ' & " and src_cat_name = " & Session("SRC_CAT_NAME") & ""
    ' & " and source_name = " & Session("SOURCE_NAME") & ""
    ' & " and Camp_name = " & Session("CAMP_NAME") & ""

    'Panel Date Start
    index = PGE.Additem
    PGE.UpdateSQL index, "Update panels set panel_date_start="
    PGE.Where index, WhereClause

    'Panel Type

```



```
'
index = PGE.AddItem
PGE.UpdateSQL index, "Update Panels set Panel_type ="
PGE.Where index, WhereClause

'Panel SubType
index = PGE.AddItem
PGE.UpdateSQL index, "Update Panels set panel_subtype="
PGE.Where index, WhereClause

'Test Type
index = PGE.AddItem
PGE.UpdateSQL index, "Update Panels set test_type="
PGE.where index, whereclause

'Package Type
index = PGE.additem
PGE.UpdateSQL index, "Update Panels set panel_pkg ="
PGE.where index, whereclause

'Expense Set
index= PGE.AddItem
PGE.UpdateSQL index, "Update Panels set e_set_name ="
PGE.Where index, WhereClause

'Revenue Set
index = PGE.AddItem
PGE.UpdateSQL index, "Update Panels set r_set_name ="
PGE.where index, whereclause

'Newstand Marker
index = PGE.AddItem
PGE.UpdateSQL index, "Update Panels set newstand_sale_marker="
PGE.where index, Whereclause

'Do it
PGE.PostGroupEditsByQuery
'...and ends here

end if

%>

<html>
<head>
<title>Admin6a.asp</title>

</head>
<% LorieScripts %>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

<table width="640" border="0" cellpadding="2">

<% prevSelections (5) %>
<form method="post" action="<%Response.Write l_this_file %>" id="form2" name="form2">
<td width="40"></td><td></td><br>
<td width="60">

<%

strSQL = "select PANEL_NAME from PANELS where " _
& " grp_name = " & Session("GRP_NAME") & "" _
& " and PRODUCT_ID=" & Session("PRODUCT_ID") & "" _
& " and src_cat_name = " & Session("SRC_CAT_NAME") & "" _
```



```
& " and source_name = "" & Session("SOURCE_NAME") & "" _
& " and Camp_name = "" & Session("CAMP_NAME") & ""

objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

showQueryListBox objRST, "cmb", 5, "SELECT PANEL",-1
objRST.close
%>
<input type="IMAGE" name="go" value="go" src="../graphics/go1222.gif"
WIDTH="33" HEIGHT="16" border=0>

</p>

</td>
</form>
<% ' setupTableListBox l_this_file %>
</tr>

</table>
</form>

<%
    l_editProc = "editPanel.asp"
    l_addProc = "addPanel.asp"

myQ = "select PANEL_NAME AS 'PANEL NAME', " _
    & " PANEL_DATE_START as 'PANEL DATE START', " _
    & " PANEL_TYPE as 'PANEL TYPE', " _
    & " PANEL_SUBTYPE as 'SUBTYPE', " _
    & " TEST_TYPE as 'TEST TYPE', " _
    & " PANEL_PKG as 'PACKAGE', " _
    & " E_SET_NAME as 'EXPENSE SET', " _
    & " R_SET_NAME as 'REVENUE SET', " _
    & " REC_ID from PANELS where " _
    & " grp_name = "" & Session("GRP_NAME") & "" _
    & " and PRODUCT_ID = "" & Session("PRODUCT_ID") & "" _
    & " and src_cat_name = "" & Session("SRC_CAT_NAME") & "" _
    & " and source_name = "" & Session("SOURCE_NAME") & "" _
    & " and Camp_name = "" & Session("CAMP_NAME") & "" _
    & " Order by SOURCE_NAME "
l_name = "Pannels"
set myObjEdit = New ObjEdit
%> <%

'Group Edit info
set GE = new objGroupEdit
GE.init
GE.action = "admin6a.asp"
GE.RowCount = 1

'Group Edit Show/Hide button
if Request.form("SHOWGROUPEdit") > "" then
    %> <form method="post" action="" <%Response.Write l_this_file %>" id="hidegeform"
name="hidegeform">
        <table>
            <tr>
                <td width="20">&nbsp;</td>
                <td><input type="submit" value="Hide Group Edit"
name="HIDEGROUPEdit" action="form.submit"></td>
            </tr>
        </table>
    </form>
    <%
        GE.Show = True
    else
```

```

%> <form method="post" action=" <%Response.Write l_this_file %>" id="showgeform"
name="showgeform">
    <table>
        <tr>
            <td width="20">&nbsp;</td>
            <td><input type="submit" value="Show Group Edit"
name="SHOWGROUPEdit" action="form.submit"></td>
        </tr>
    </table>
</form>
<br> <form method="POST" action="<%=action%>" name="Group_Edit_Form" cellpadding="3"
cellspacing="3">
    <%
    GE.Show = False
    end if

    'Panel Name
    index = GE.AddItem
    GE.fieldType index, "LABEL"
    GE.ColumnHeader index, "PANEL NAME"

    'Panel Date Start
    index = GE.AddItem
    GE.fieldType index, "TEXT"
    GE.ColumnHeader index, "PANEL DATE START"

    'Panel Type
    index = GE.AddItem
    GE.fieldType index, "SELECT"
    GE.ColumnHeader index, "PANEL TYPE"
    GE strSQL index, "SELECT panel_type_name FROM corp_panel_type ORDER BY panel_type_name"

    'Panel SubType
    index = GE.AddItem
    GE.fieldtype index, "SELECT"
    GE.ColumnHeader index, "SUBTYPE"
    GE strSQL index, "SELECT panel_sub_type_name from corp_panel_sub_type order by panel_sub_type_name"

    'Test type
    index = GE.AddItem
    GE.fieldtype index, "SELECT"
    GE.ColumnHeader index, "TEST TYPE"
    GE strSQL index, "SELECT panel_test_type_name from corp_panel_test_type order by panel_test_type_name"

    'Package Type
    index = GE.AddItem
    GE.fieldtype index, "SELECT"
    GE.ColumnHeader index, "PACKAGE"
    GE strSQL index, "SELECT package_name from prod_panel_package where product_id = " &
Session("PRODUCT_ID") & " order by package_name"

    'Expense Set
    index = GE.AddItem
    GE.fieldtype index, "SELECT"
    GE.ColumnHeader index, "EXPENSE SET"
    GE strSQL index, "SELECT e_set_name from expense_sets where product_id = " & Session("PRODUCT_ID")
& " order by e_set_name"

    'Revenue Set
    index = GE.AddItem
    GE.fieldType index, "SELECT"
    GE.ColumnHeader index, "REVENUE SET"
    GE strSQL index, "SELECT r_set_name from revenue_sets where product_id = " & Session("PRODUCT_ID")
& " order by r_set_name"

    'Newstand Marker
    index = GE.AddItem
    GE.fieldType index, "SELECTLOV"

```

```
'GE.ColumnHeader index, "NEWSTAND MARKER"
'GE strSQL index, "N,Y"

'Page
GE.MS_Page = 1
'GE.MS_Table = "panels"
'GE.MS_WhereClause = "where grp_name = " & Session("GRP_NAME") & ""
'                                     & " and PRODUCT_ID=" & Session("PRODUCT_ID") & ""
'                                     & " and src_cat_name = " & Session("SRC_CAT_NAME") & ""
-                                     & " and source_name = " & Session("SOURCE_NAME") & ""
-                                     & " and Camp_name = " & Session("CAMP_NAME") & ""
'GE.MS_OrderBy = " Order by SOURCE_NAME "

'Display Table
GE.MS_Page = 1
GE.DisplayEditTable

myObjEdit.Init l_Name, l_editProc, l_addProc, l_this_file
myObjEdit.defineComplex(10)
myObjEdit.setQuery(myQ)
myObjEdit.SelectTable

%>

</body>
</html>
```



adminindex.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="99CCFF" background="../graphics/adminhorzslice.gif" text="#000000" link="6699CC" marginwidth="0"
marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action id="form1" name="form1">
<table width="660" border="0" cellpadding="10" cellspacing="10">
<tr>
<td width="162"> <br>
<form METHOD="POST" ACTION="AdminIndex.asp" id="form2" name="form2">
<%
'dim src
'src = Request.Form("Go")
dim v_state, v_tmp, v_name
dim strTable, strField, strWhere

v_state = Request.Form("state")

if v_tmp <> "0" and v_tmp > " " then
if v_state = "PRODUCT GROUP" THEN
v_state = "PRODUCT"
Session("TABLE") = " CORP_INFO "
Session("FIELD") = " PRODUCT_ID, TITLE_NAME"
Session("WHERE") = " GRP_NAME = " & v_tmp & ""
' Response.Write "<input type='Hidden' name='GRP' value=" & v_tmp & "> "
elseif v_state = "PRODUCT" THEN
v_state = "SOURCE CATEGORY"
Session("TABLE") = " SRC_CATEGORY "
Session("FIELD") = " SRC_CAT_NAME "
Session("WHERE") = " PRODUCT_ID = " & v_tmp & " "
Session("PRODUCT_ID") = v_tmp
end if
else
' default case
v_state = "PRODUCT GROUP"
v_name = "GRP"
Session("TABLE") = " CORP_INFO "
Session("FIELD") = " distinct GRP_NAME "
Session("WHERE") = " "

'ShowRecords objRST. "HI dude"

end if
strTable = Session("TABLE")
strField = Session("FIELD")
strWhere = Session("WHERE")
' Response.Write (strTable & strField & strWhere)

strSQL = "select " & strField & " from " & strTable & " where " & strWhere
if strWhere > " " THEN
strSQL = strSQL & " where " & strWhere
end if
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showListBox objRST, "cmb", false, "SELECT " & v_state, false
objRST.close
'Response.Write "<input type='Hidden' name=" & v_name & " value=" & v_state & "> "
```



```
%>
</td>
<td rowspan="6" width="468" valign="top">
  <div align="center">
    <p>&nbsp;</p>
    <p>&nbsp;</p>
  </div>
</td>
</tr>
<tr>
  <td width="162">
    <div align="right">

      <input type="image" src="../../graphics/smallgo.gif" width="83" height="30" name="Go" Alt="Go">
    </div>
  </td>
</tr>
<%
v_grp = Request.Form("GRP")
if v_state = "PRODUCT_GROUP" THEN

%>

  <div align="right">
  </div>
</td>
</tr>
<tr>
  <td width="162">
    <table width="190" border="0" cellspacing="0" cellpadding="0" align="center">
      <tr>
        <td height="25">
          <div align="right"></div>
        </td>
        <td height="25">
          <div align="left"></div>
        </td>
      </tr>
      <tr>
        <td>
          <div align="right"><font size="4" face="Verdana, Arial, Helvetica, sans-serif">Product
            Group: </font></div>
        </td>
        <td><font size="4" face="Verdana, Arial, Helvetica, sans-serif"><b>&lt;variable</b></font>&gt;</td>
      </tr>
      <tr>
        <td>
          <div align="right"><font size="4" face="Verdana, Arial, Helvetica, sans-serif">Product:</font>
          </div>
        </td>
        <td><font size="4" face="Verdana, Arial, Helvetica, sans-serif"><b>&lt;variable</b></font>&gt;</td>
      </tr>
      <tr>
        <td>
          <div align="right"><font size="4" face="Verdana, Arial, Helvetica, sans-serif">Source
            Category:</font> </div>
        </td>
        <td><font size="4" face="Verdana, Arial, Helvetica, sans-serif"><b>&lt;variable</b></font>&gt;</td>
      </tr>
      <tr>
        <td>
          <div align="right"><font size="4" face="Verdana, Arial, Helvetica, sans-serif">Source:</font>
          </div>
        </td>
        <td><font size="4" face="Verdana, Arial, Helvetica, sans-serif"><b>&lt;variable</b></font>&gt;</td>
      </tr>
    </table>
  </td>
</tr>
```



```
<td>
  <div align="right"><font size="4" face="Verdana, Arial, Helvetica, sans-serif">Campaign:</font>
</div>
</td>
<td><font size="4" face="Verdana, Arial, Helvetica, sans-serif"><b>&lt;variable</b></font>&gt;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
</table>
</td>
</tr>
<tr>
<td width="162">
<p></p>
<p>
  <select name="SELECT ONE">
    <option>SELECT ONE</option>
    <option>&lt;Variable&gt;</option>
    <option>&lt;Variable&gt;</option>
  </select>
</p>
</td>
</tr>
<tr>
<td width="162">
  <div align="right">
  </div>
</td>
</tr>
<% end if %>

</table>
</form>
</body>
</html>
```



adminindexlogon.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<%
dim strName
dim strPass
strName = Request.ServerVariables("AUTH_USER")
strPass = Request.ServerVariables("AUTH_PASSWORD")
Session("U") = strName
Session("P") = strPass

%>

<html>
<head>
<title>IMT Network</title>

</head>

<frameset rows="97,*" noresize cols="*" frameborder="NO" border="0" framespacing="0">
<frame src="adminnavbar.asp" name="navigation" scrolling="NO" noresize frameborder="NO">
<frame src="welcome.asp" name="main" scrolling="YES" frameborder="NO">
</frameset>
<noframes><body bgcolor="#FFFFFF">

</body></noframes>
</html>
```

adminkeys.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%
l_debug = 1 ' set to 1 if you want to debug this page

if (Request.Form("display")="2" _
    and Request.Form("setupTab") <> "0" _
    and Request.Form("setupTab") > " ") then
    Response.Redirect(Request.Form("setupTab"))
end if

if Request("add")="1" then
    l_AvailCount = Request.Form("AvailCount")
    if l_debug = 1 then
        Response.Write l_availcount & "<br>"
    end if
    l_AvailCount = clng(l_AvailCount)
    for j = 1 to l_availcount
        l_val = Request.Form("p_check" & j)
        ' Response.Write j & "Work damn it" & l_val & "<br>"
        if l_val > " " then

            my_rec_id = clng(l_val)
            objCMD.ActiveConnection = strSQLOLEDB
            objCMD.CommandText = "sp_Add_Keys"
            objCMD.CommandType = adCmdStoredProc
            if l_debug = 1 then
                Response.write "rec_id " & rec_id & "<br>"
                Response.write "Session('SRC_CAT_NAME') " & Session("SRC_CAT_NAME") & "<br>"
                Response.write "Session('SRC_INDICATOR') " & Session("SRC_INDICATOR") & "<br>"
                Response.write "Session('CAMP_NAME') " & Session("CAMP_NAME") & "<br>"
                Response.write "Session('PANEL_NAME') " & Session("PANEL_NAME") & "<br>"
            end if
            objCMD.Parameters.Refresh
            objCMD.Parameters(1).Value = my_rec_id
            objCMD.Parameters(2).Value = Session("SRC_CAT_NAME")
            objCMD.Parameters(3).Value = Session("SOURCE_NAME")
            objCMD.Parameters(4).Value = Session("SRC_INDICATOR")
            objCMD.Parameters(5).Value = Session("CAMP_NAME")
            objCMD.Parameters(6).Value = Session("PANEL_NAME")
            if l_debug <> 1 then

                Set TempNonSessionVar = objCMD.Execute()
            else
                RESPONSE.write "would have executed here<br>"
            end if
            ' Response.Write my_rec_id & "<br>"
        end if
    next
end if

dim l_pos(10)
'strSQL = " select full_promo_key + '-' + ltrim(str(mail_qty)) + '-' _
'& " + convert(varchar(12),dt_valid_to) MY_KEY, REC_ID " _
strSQL = " select full_promo_key MY_KEY, REC_ID " _
& " from avail_keys " _
& " where Product_id = " & session("PRODUCT_ID")
'& " and source_identifier = " & session("SRC_INDICATOR") & " "
ORDERby = " Order by "
```

```

for i = 1 to 10
  l_pos(i) = Request("KeyPosition" & i)
  if l_pos(i) > " " then
    'SQL
    strSQL = strSQL & " and POSITION" & i _
    & " = " & Request("KeyPosition" & i) & " "
    'Order by clause

    runMe = true

  end if
  OrderBy = OrderBy & " Position" & i & ", "
next
if (Request.Form("runMe") = "1") then
  runMe = true
end if
Orderby = Orderby & " Full_Promo_key "
strSQL = strSQL & " " & OrderBy
'Response.Write strSQL & "<br>"
if (Request.Form("display") < " ") or (l_pos(1) < " ") then
  l_pos(1) = Session("SRC_INDICATOR")
end if
'Response.Write " <BR> " & strSQL & "<BR>"
%>

<html>
<head>
<title>IMT Network</title>

<script language="JavaScript">
<!--
function MM_swapImgRestore() { //v2.0
  if (document.MM_swapImgData != null)
    for (var i=0; i<(document.MM_swapImgData.length-1); i+=2)
      document.MM_swapImgData[i].src = document.MM_swapImgData[i+1];
}

function MM_preloadImages() { //v2.0
  if (document.images) {
    var imgFiles = MM_preloadImages.arguments;
    if (document.preloadArray==null) document.preloadArray = new Array();
    var i = document.preloadArray.length;
    with (document) for (var j=0; j<imgFiles.length; j++) if (imgFiles[j].charAt(0)!="#"){
      preloadArray[i] = new Image;
      preloadArray[i++].src = imgFiles[j];
    }
  }
}

function MM_swapImage() { //v2.0
  var i,j=0,objStr,obj.swapArray=new Array,oldArray=document.MM_swapImgData;
  for (i=0; i < (MM_swapImage.arguments.length-2); i+=3) {
    objStr = MM_swapImage.arguments[(navigator.appName == "Netscape"?i:i+1)];
    if ((objStr.indexOf("document.layers['"])=0 && document.layers==null) ||
      (objStr.indexOf("document.all['"])=0 && document.all ==null))
      objStr = 'document'+objStr.substr(objStr.lastIndexOf(' '),objStr.length);
    obj = eval(objStr);
    if (obj != null) {
      swapArray[j++] = obj;
      swapArray[j++] = (oldArray==null || oldArray[j-1]!==obj)?obj.src:oldArray[j];
      obj.src = MM_swapImage.arguments[i+2];
    }
  }
  document.MM_swapImgData = swapArray; //used for restore
}
//-->
</script>

```



```
</head>
<body>
```

```
<table width="1253" border="0" cellspacing="10" cellpadding="0">
  <tr valign="top">
    <td width="195" height="23" valign="top">
      <table>
        <tr>
          <td>
            
          </td>
        </tr>
      </table>
    </td>
    <td width="195" rowspan="2" height="30" valign="top" align="left">
      <div align="left"></div>
      <table width="180" border="0" cellspacing="1" cellpadding="0" align="left">
        <tr>
          <% i = 0
            for i = 1 to 10 %>
            <td>
              <div align="center">
                <font size="-4" face="Verdana, Arial, Helvetica, sans-serif">
                  <b><% Response.Write i %>
                </b></font></div>
              </td>
              <%next %>
            </tr>
            <tr>
              <% for i = 1 to 10 %>
              <td>
                <div align="center">
                  <input type="text" name="KeyPosition<% Response.Write i %>" size="1" maxlength="2" value="<% Response.Write
l_pos(i) %>">
                </div>
              </td>
              <%next %>
            </tr>
          </table>
        </td>
        <tr>
          <td>
            <div align="right">
              <input type="image" src="../../graphics/smallgo.gif" width="83" height="30" name="Go" Alt="Go">
            </div>
          </td>
        </tr>
        <tr>
          <td width="195">
            <% sessionInfo (6) %>
            <% accesSetupTab %>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</table>
</form>
```

```

<% ' it all starts here %>

<td rowspan="7" width="1556" valign="top">
<div align="center">
<p><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">
</font></b></font></p>
<div align="left">
<table width="1548" border="0" cellspacing="3" cellpadding="2">

<% IF RUNME THEN
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
if (objRST.EOF = true) then
objRst.close
%>
<tr>
<td VALIGN="TOP">
<table>
<tr>
<td width="24" height="25" align="center">
<FONT COLOR="RED">
No Available Keys Meet this Criteria </FONT></td>
</tr></table>
<%else
%>
<tr>
<td VALIGN="TOP">
<table> <% 'Available Keys Table %>
<tr>

<td width="24" height="25" align="center">&nbsp;</td>
<td width="24" height="25" align="center">

<form method="post" name="myForm">
<input type="Submit" id="Submit1" name="Submit1" value="Assign Keys">
<input type="button" value="Select All" onClick="checkAll()">
<input type="button" value="Unselect All" onClick="unCheckAll()">
<input type="hidden" value="1" name="add">
<input type="hidden" value="1" name="runMe">
<% for i = 1 to 10

if l_pos(i) > " " then %>
<input type="hidden" value="<%Response.Write l_pos(i) %>"
name="KeyPosition<%Response.Write i %>">

<%
end if

next %>
<input type="hidden" value="1" name="runMe">
</td>
</tr>
<tr>
<td width="24" height="25" align="center">&nbsp;</td>
<td width="140" height="25">&nbsp;</td>
</tr><tr>
<td width="24" height="25" align="center">
<div align="center">
<br>
</div>
</td>
<td width="140" height="25" bgcolor="003366">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b>
<font size="1">
AVAILABLE
KEYS
</font></b></font></div>

```



```

</td>
</tr>

<%
    dim i ' count variable
    i=0
    Do While ObjRST.EOF = False
        i = i + 1
    <%>

<tr>
<td width="24" align="center">
<div align="center">
    <input type="checkbox" name="p_check"<%Response.write i%>" value="<% Response.write ObjRST.Fields("REC_ID")
%>">
    </div>
</td>
<td width="140" bgcolor="FFFFFF">
<font size="-4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("MY_KEY") %>
</font></td>
</tr>
<%
    ObjRst.moveNEXT
    loop
    objRst.close %>
    <input type="Hidden" name="AvailCount" value="<%Response.write i %>">

<SCRIPT LANGUAGE="JavaScript">

    <!-- Begin
    function checkAll() {
    for (var j = 1; j <= <% Response.write i %>; j++) {
    box = eval("document.myForm.p_check" + j);
    if (box.checked == false) box.checked = true;
    }
    }

    function unCheckAll() {
    for (var j = 1; j <= <% Response.write i %>; j++) {
    box = eval("document.myForm.p_check" + j);
    if (box.checked == true) box.checked = false;
    }
    }

    // End -->
</script>

</table>

</td>
</form>

<% end if

end if %>
<td VALIGN="TOP">
<table> <% 'Assigned Keys Table' %>
<tr>

<td width="75" height="25">&nbsp;</td>
<td width="25" height="25">&nbsp;</td>
<td height="25" colspan="8">
<div align="left">

```



```
color="#FFFFFF"><b>
                                <font face="Verdana, Arial, Helvetica, sans-serif"
                                <font size="1"><% RResponse.write Session("PANEL_NAME")
                                </font></b></font></div>
                                </td>
                                </tr>
                                <tr>
                                <td width="75" height="25">&nbsp;</td>
                                <td width="25" height="25">&nbsp;</td>
                                <td bgcolor="003366" height="25" width="140">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1">PROMOTION
                                KEY </font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="210" align="center">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1">KEY
                                DESCRIPTION </font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="77">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1">MAIL
                                QTY </font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="119">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1"><nobr>MAIL
                                QTY OVERRIDE</font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="238">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1">LIST</font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="56">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1">LIST
                                CPM</font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="175">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1"><nobr>EXPENSE
                                SET </font></b></font></div>
                                </td>
                                <td bgcolor="003366" height="25" width="175">
                                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
                                color="#FFFFFF"><b><font size="1"><nobr>REVENUE
                                SET </font></b></font></div>
                                </td>
                                </tr>
                                </%>

StrSQL = "select FULL_PROMO_KEY, P_KEY_DESCR, KEY_MAIL_QTY, " _
        & " MAIL_QTY_OVERRIDE, LIST_NAME, LIST_COST_ROLLOUT_CPM, E_SET_NAME, " _
        & " R_SET_NAME, KEYS.REC_ID REC_ID " _
        & " from keys,list_names where keys.product_id=" & Session("PRODUCT_ID") _
        & " and keys.camp_name = " & Session("CAMP_NAME") & " " _
        & " and keys.source_name = " & Session("SOURCE_NAME") & " " _
        & " and keys.panel_name=" & Session("PANEL_NAME") & " " _
        & " and keys.src_cat_name=" & Session("SRC_CAT_NAME") & " " _
        & " and keys.product_id = list_names.product_id " _
        & " and keys.list_id = list_names.rec_id " _
        & " Order by full_promo_key "
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
count variable
i=0
Do While ObjRST.EOF = False
i = i + 1
```

```
%>
```

```
<tr>
```

```

    <% if (i = 1) then    %>
    <td width="75" rowspan="3" align="center">
    <div align="center"></div>
    <div align="center"><nobr>
    <font face="Verdana, Arial, Helvetica, sans-serif"><font size="1">
    </font></font></div>
    <div align="center">
    <br>
    </font>
    </div>
    </td>
    <% elseif (i= 2 or i = 3) then %>

```

```

    <% else %>
    <td> &nbsp; </td>

```

```
    <% end if %>
```

```
<td width="25">
```

```

    <div align="right">
    <a href="editKeys.asp?Z=<%Response.write ObjRST.Fields("REC_ID") %>">
    
    </a>

```

```
    </div>
```

```
</td>
```

```

<td width="140" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("FULL_PROMO_KEY") %>
</font></td>

```

```

<td width="210" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("P_KEY_DESCR") %>
</font></td>

```

```

<td width="77" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("KEY_MAIL_QTY") %>
</font></td>

```

```

<td width="119" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("MAIL_QTY_OVERRIDE") %>
</font></td>

```

```

<td width="238" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("LIST_NAME") %>
</font>

```

```
</td>
```

```

<td width="56" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("LIST_COST_ROLLOUT_CPM") %>
</font>

```

```
</td>
```

```

<td width="175" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("E_SET_NAME") %>
</font>

```

```
</td>
```

```

<td width="175" bgcolor="#FFFFFF" height="15">
<font size="4" face="Verdana, Arial, Helvetica, sans-serif">
<% Response.write ObjRST.Fields("R_SET_NAME") %>
</font>

```

```
</td>
```



```
</tr>
<%
objRST.moveNext
loop %>
</table>
</td>
</tr>

</table>
</div>
<p>&nbsp;</p>
</div>
</td>
<% ' it all ends here %>
</tr>

</table>
</form>
</body>

</html>
```

adminkeysleft.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/keys.asp" -->
<% checkSessionVars(6) %>

<html>
<head>
<script>
function refresh(){
    var kevin = parent.frames.Right ;
    kevin.location.href = "adminKeysRight.asp" ;
}

</script>
<title>IMT Network</title>

</head>

<%
l_debug = 0
l_THIS_FILE = "adminKeysLeft.asp"
l_AvailCount = Request.Form("AVAILCOUNT")
' Response.Write "availCount = " & l_availcount & "<br>"
'on error resume next

if Request.Form("ADD")="1" then
' Response.Write l_availcount & "<br>"
l_AvailCount = Request.Form("AvailCount")
' Response.Write l_availcount & "<br>"
l_AvailCount = clng(l_AvailCount)
server.ScriptTimeout = 5000000
for j = 1 to l_availcount
    l_val = Request.Form("p_check" & j)
    ' if l_debug = 1 then

    ' Response.Write j & "Work damn it" & l_val & "<br>"
    ' end if
    if l_val > " " then

        my_rec_id = clng(l_val)
        objCMD.ActiveConnection = strSQLOLEDB_alt
        objCMD.CommandText = "sp_Add_Keys"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        if l_debug = 1 then
            Response.Write my_rec_id & "<br>"
            Response.Write Session("SRC_CAT_NAME") & "<br>"
            Response.Write Session("SOURCE_NAME") & "<br>"
            Response.Write Session("SRC_INDICATOR") & "<br>"
            Response.Write Session("CAMP_NAME") & "<br>"
            Response.Write Session("PANEL_NAME") & "<br>"
        end if

        objCMD.Parameters(1).Value = my_rec_id
        objCMD.Parameters(2).Value = Session("SRC_CAT_NAME")
        objCMD.Parameters(3).Value = Session("SOURCE_NAME")
        objCMD.Parameters(4).Value = Session("SRC_INDICATOR")
        objCMD.Parameters(5).Value = Session("CAMP_NAME")
        objCMD.Parameters(6).Value = Session("PANEL_NAME")
        if l_debug <> 1 then
            Set TempNonSessionVar = objCMD.Execute()
        else

```



```
Response.write "I would have executed here, pal,<br> but you're debugging. Turn
debug off in the source code to resume normal operations<br>";
end if

end if
next

end if

%>

<% dim l_query
dim l_pos(10)
dim l_display
l_display = Request.Form("DISPLAY")
l_query = " select full_promo_key MY_KEY, REC_ID " _
& " from avail_keys " _
& " where Product_id = " & session("PRODUCT_ID")

l_query = " select full_promo_key MY_KEY, REC_ID " _
& " from avail_keys " _
& " where Product_id = " & session("PRODUCT_ID")
'& " and source_identifier = " & session("SRC_INDICATOR") & " "
ORderby = " Order by "

for i = 1 to 10
    l_pos(i) = trim(Request.form("KeyPosition" & i))
    if l_pos(i) > " " then
        l_query = l_query & " and POSITION" & i & " = " & trim(Request.Form("KeyPosition" & i)) & " "
        l_query = l_query & " and substring(full_promo_key," & i & ",1) = " & l_pos(i) & " "
        runMe = true
    end if
    ' OrderBy = OrderBy & " Position" & i & ", "
next

if (Request.Form("runMe") = "1" ) then
    runMe = true
end if

Orderby = ORderby & " Full_Promo_key "

if (trim(l_pos(1)) < " ")then
    l_pos(1) = Session("SRC_INDICATOR")
    l_query = l_query & " and POSITION1" _
    & " = " & trim(l_pos(1)) & " "
end if

l_query = l_query & " " & OrderBy

%>

<% if Request("add")="1" then %>

<body bgcolor="#99CCFF" vlink="000000" link="003366"
marginwidth="0" marginheight="0" leftmargin="0" topmargin="0"
onLoad="refresh();">

<% else %>

<body bgcolor="#99CCFF" vlink="000000" link="003366"
marginwidth="0" marginheight="0" leftmargin="0" topmargin="0" >
```



```
<%  
end if  
%>
```

```
<table width="185" cellpadding="2">  
  <tr>  
    <td colspan="3" height="39" width="152">  
      <% keySearchBoxes l_pos,"adminKeysLeft.asp", false %>  
      <tr valign="top" align="right">  
        <td colspan="4">  
          <div align="left"><br>  
          </div>  
          <div align="left"></div>  
        <p>  
        <form method="post" action="<% Response.Write L_THIS_FILE %>" name="myForm" id=form1>  
  
        <% for i = 1 to 10 %>  
          <input type="HIDDEN"  
            name="KeyPosition<% Response.Write i %>"  
            value="<% Response.write l_pos(i) %>">  
          <% next  
          if l_display = "1" then %>  
            <input type="IMAGE" src=" ../graphics/assignkeys1222.gif" width="91" height="28" border="0">  
          <% end if %>  
        </p>  
      </td>  
    </tr>  
  
    <tr valign="top">  
      <td colspan="4" align="left"> <br>  
  
      <input type="HIDDEN" name="ADD" value="1">  
      <input type="HIDDEN" name="DISPLAY" value="1">  
      <%  
      if l_display = "1" then  
        'Response.Write l_query & "<BR>"  
        AvailKeysTable(l_query)  
      end if %>  
  
    </td>  
  </tr>  
</table>  
</form>  
</body>  
</html>
```

adminkeysright.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/keys.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/adoutils.asp" -->
<!-- #include file="../support/objEdit.asp" -->

<%

L_THIS_FILE = "adminKeysRight.asp"
response.expires = 0
response.expiresabsolute = Now() - 1
response.addHeader "pragma","no-cache"
response.addHeader "cache-control","private"
Response.CacheControl = "no-cache"
if Request.Form("ACTION") = "PANEL" then
    l_panel = Request.Form("cmb")
    Session("PANEL_NAME") = l_panel
end if

'Dim vars

'Common
dim recordcount
dim checked_records
dim i
dim WhereClause
dim recordidlist

'Edit vars
dim MSE

'Group Edit vars
dim PGE
dim index
dim recordid

'Edit
if Request.form("editbutton") > " " then
    'Setup first page, then let edit handle the rest

    'init obj
    set MSE = new objGroupEdit

    'Get records
    recordcount = Request.form("p_RecCount")
    toomanyrecords = 0
    'response.write("There are " & cstr(recordcount) & " records <br>")

    checked_records = 0
    for i = 1 to recordcount
        recordid = Request.form("P_check" & cstr(i))
        if recordid > " " and len(recordidlist) > 0 then
            checked_records = checked_records + 1
            if checked_records > 19 then
                toomanyrecords = 1
            else
                recordidlist = recordidlist + "," + cstr(recordid)
            end if
        elseif recordid > " " then ' don't add comma first time
            recordidlist = cstr(recordid)
        end if
    next i
```



```

Session("RECORDIDLIST") = recordidlist
session("page") = 1
Session("toomanyrecords") = toomanyrecords
if recordidlist > " " then response.redirect("ms_editkey.asp?pg=1")
end if

'Post/Delete Group Edit Info
if Request.form("GroupEdit") > " " or request.form("deletebutton") > " " then

    'response.write ("Hello World!@!!!!!!")
    response.write("&nbsp;")
    'init obj
    set PGE = new objPostGroupEdit
    PGE.init

    'Get records
    recordcount = Request.form("p_RecCount")
    'response.write("<br>There are " & cstr(recordcount) & " records <br>")
    for i = 1 to recordcount
        recordid = Request.form("P_check" & cstr(i))
        if recordid > " " then PGE.AddRecord recordid
        'response.write(recordid & "<br>")
    next i

    'Set Stored Procedure info
    PGE.ClassType = "Group Edit"
    PGE.SP = "sp_edit_key"
    PGE.TableName = "keys"
    if Request.form("GroupEdit") > " " then PGE.IsDelete = 0
    if request.form("deletebutton") > " " then PGE.IsDelete = 1
    PGE.AddParam "p_key_descr"
    PGE.AddParam "mail_qty_override"
    PGE.AddParam "newstand_sale"
    PGE.AddParam "newstand_sale_marker"
    PGE.AddParam "list_cat_name"
    PGE.AddParam "list_name"
    PGE.AddParam "list_segment_name"
    PGE.AddParam "list_cost_actual_cpm"
    PGE.AddParam "list_cost_basis"
    PGE.AddParam "list_cost_factor"
    PGE.AddParam "merge_purge_qty"
    PGE.AddParam "e_set_name"
    PGE.AddParam "r_set_name"
    PGE.AddParam "panel_pkg"
    PGE.AddParam "subtotal_flag"
    PGE.AddParam "optional_effort_id"
    PGE.AddParam "rec_id"
    PGE.AddParam "IsDelete"
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_edit_key"
    objCMD.CommandType = adCmdStoredProc
    objCMD.parameters.refresh
    PGE.PostGroupEditsBySP
end if
%>
<html>
<head>
<META HTTP-EQUIV="Expires" CONTENT="Mon, 01 Jan 1990 12:00:00 GMT">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<title>IMT Network</title>
<% LorieScripts %>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">

```



```
onLoad="MM_preloadImages('./graphics/edittablero1222.gif','#945899226033');MM_preloadImages('./graphics/edittablero1222.gif',
'#945899269583')">
<br><br>
```

```
<form method="post" action="<% Response.Write L_THIS_FILE %>" name=form1 id=form1>
<input type="HIDDEN" name="ACTION" VALUE="PANEL">
<table width="2500" cellpadding="2">
<% prevSelections (6) %>
```

```
<tr valign="top">
<td><font size="-4" face="Verdana, Arial, Helvetica, sans-serif">
Change Panel<br>
</font>
<%
```

```
        strSQL = "select PANEL_NAME from PANELS where " _
        & " grp_name = " & Session("GRP_NAME") & " " _
        & " and PRODUCT_ID=" & Session("PRODUCT_ID") & " " _
        & " and src_cat_name = " & Session("SRC_CAT_NAME") & " " _
        & " and source_name = " & Session("SOURCE_NAME") & " " _
        & " and Camp_name = " & Session("CAMP_NAME") & " "
```

```
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        showQueryListBox objRST, "cmb",3, "SELECT PANEL", Session("PANEL_NAME")
objRST.close
%>
<input type="IMAGE" name="go" value="go" src="./graphics/go1222.gif"
WIDTH="33" HEIGHT="16" BORDER=0>
```

```
</td>
</tr>
</form>
</table>
```

```
<%
'Group Edit info
```

```
        set GE = new objGroupEdit
        GE.init
        GE.action = "adminkeysright.asp"
        GE.RowCount = 1
```

```
        'Group Edit Show/Hide button
        if Request.form("SHOWGROUPEdit") > " " then
            %>        <form method="post" action="<%Response.Write l_this_file %>" id="hidegeform"
name="hidegeform">
```

```
                <table>
                    <tr>
                        <td width="20">&nbsp;</td>
                        <td><input type="submit" value="Hide Group Edit"
name="HIDEGROUPEdit" action="form.submit"></td>
                    </tr>
                </table>
            </form>
```

```
            <%
            GE.Show = True
        else
            %> <form method="post" action="<%Response.Write l_this_file %>" id="showgeform"
name="showgeform">
                <table>
                    <tr>
                        <td width="20">&nbsp;</td>
                        <td><input type="submit" value="Show Group Edit"
name="SHOWGROUPEdit" action="form.submit"></td>
                    </tr>
                </table>
```



```
</form>
<br> <form method="POST" action="<%=action%>" name="Group_Edit_Form" cellpadding="3"
cellspacing="3">
  <%
    GE.Show = False
  end if

  'Key
  index = GE.AddItem
  'GE.fieldType index, "NOTAPARAMETER"
  'GE.ColumnHeader index, "KEY"

  'Key Description
  index = GE.AddItem
  GE.fieldType index, "TEXT"
  GE.ColumnHeader index, "KEY DESCRIPTION"

  'Mail Qty
  index = GE.AddItem
  'GE.fieldType index, "LABEL"
  'GE.ColumnHeader index, "MAIL QTY"

  'Mail Qty OverRide / Draw
  index = GE.AddItem
  GE.fieldtype index, "TEXT"
  GE.ColumnHeader index, "MAIL QTY OVERRIDE / DRAW"

  'Newsstand Sale
  index = GE.AddItem
  GE.fieldtype index, "TEXT"
  GE.ColumnHeader index, "NEWSSTAND SALE"

  'Response Based on Sale or Draw (Newstand Sale Marker)
  index = GE.AddItem
  GE.fieldtype index, "SELECTLOV"
  GE.ColumnHeader index, "RESPONSE BASED ON SALE (Y), DRAW (N)"
  GE.strSQL index, "N,Y"

  'List Category
  index = GE.AddItem
  GE.fieldtype index, "SELECT"
  GE.ColumnHeader index, "LIST CATEGORY"
  GE.strSQL index, "SELECT list_cat_name from list_cats where product_id = " & Session("PRODUCT_ID") &
" order by list_cat_name"

  'List Name
  index = GE.AddItem
  GE.fieldtype index, "SELECT"
  GE.ColumnHeader index, "LIST NAME"
  GE.strSQL index, "SELECT list_name from list_names where product_id = " & Session("PRODUCT_ID") & "
order by list_name"

  'List Segment
  index = GE.AddItem
  GE.fieldtype index, "SELECT"
  GE.ColumnHeader index, "LIST SEGMENT"
  GE.strSQL index, "SELECT list_segment_name from list_segments where product_id = " &
Session("PRODUCT_ID") & " order by list_segment_name"

  'List CPM
  index = GE.AddItem
  GE.fieldType index, "TEXT"
  GE.ColumnHeader index, "LIST CPM"

  'List Cost Basis
  index = GE.AddItem
  GE.fieldType index, "SELECTLOV"
  GE.ColumnHeader index, "LIST COST BASIS"
```

```

GE.strSQL index, "Mail Qty, Merge Purge Qty"

'List Cost Factor
index = GE.AddItem
GE.fieldType index, "TEXT"
GE.ColumnHeader index, "LIST COST FACTOR"

'Merge Purge Qty
index = GE.AddItem
GE.fieldType index, "TEXT"
GE.ColumnHeader index, "MERGE PURGE QTY"

'Expense Set
index = GE.AddItem
GE.fieldtype index, "SELECT"
GE.ColumnHeader index, "EXPENSE SET"
GE.strSQL index, "SELECT e_set_name from expense_sets where product_id = " & Session("PRODUCT_ID")
& " order by e_set_name"

'Revenue Set
index = GE.AddItem
GE.fieldtype index, "SELECT"
GE.ColumnHeader index, "REVENUE SET"
GE.strSQL index, "SELECT r_set_name from revenue_sets where product_id = " & Session("PRODUCT_ID")
& " order by r_set_name"

'Package
index = GE.AddItem
GE.fieldtype index, "SELECT"
GE.ColumnHeader index, "PACKAGE"
GE.strSQL index, "SELECT package_name from prod_panel_package where product_id = " &
Session("PRODUCT_ID") & " order by package_name"

'Subtotal Flag
index = GE.AddItem
GE.fieldtype index, "SELECTLOV"
GE.columnHeader index, "SUBTOTAL FLAG"
GE.strSQL index, "True,False"

'Optional Effort ID
index = GE.AddItem
GE.fieldType index, "TEXT"
GE.ColumnHeader index, "OPTIONAL EFFORT ID"

'Page
GE.MS_Page = 1
'if not(RecCount >=1) then RecCount = 1

'GE.MS_Table = "panels"
'GE.MS_WhereClause = "where grp_name = " & Session("GRP_NAME") & ""
'                                     & " and PRODUCT_ID=" & Session("PRODUCT_ID") & ""
'                                     & " and src_cat_name = " & Session("SRC_CAT_NAME") & ""
'                                     & " and source_name = " & Session("SOURCE_NAME") & ""
'                                     & " and Camp_name = " & Session("CAMP_NAME") & ""
'GE.MS_OrderBy = " Order by SOURCE_NAME "

'Display Table
GE.MS_Page = 1
GE.DisplayEditTable

%>

<table width="2500" cellpadding="2">
<tr valign="top">
<td><BR>
</td>

```



</tr>

<tr valign="top">

<td colspan="4" align="left">

<% keysTable %>

</td>

</tr>

</table>

</form>

</body>

</html>

adminsearch.asp

```
<%@ Language=VBScript %>

<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->
<!-- #include file="../support/navigation.asp" -->

<% JSImageFunctions %>

<HTML>
<HEAD>

<% DisplayIMTFrameset %>

<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE> IMT Admin Search </title>
</HEAD>
<BODY marginwidth=0 leftmargin=0 marginheight=0 topmargin=0 bgcolor="#99CCFF" vlink="#003366" link="#003366">

<% LorieScripts %>
<form method="post" action="AdminSearch.asp" id=form1 name=form1>

<%
' Response.Write SESSION("PID_PHRASE") & "<BR>"
' our local variables
p_sel = Request.Form("P_SEL")
p_grp = Request.Form("p_grp")
p_title = Request.Form("P_TITLE")
p_src_cat = Request.Form("P_SCR_CAT")
p_source = Request.Form("P_SOURCE")
p_camp = Request.Form("P_CAMP")
p_Panel = Request.Form("P_PANEL")
' if p_grp < " " then
' p_grp = "%"
' end if
if p_title < " " or p_title= "0" then
p_title = "%"
end if
if p_src_cat < " " or p_src_cat = "0" then
p_src_cat = "%"
end if
if p_Source < " " or p_Source = "0" then
p_Source = "%"
end if
if p_camp < " " then
p_camp = "%"
end if
if p_panel < " " then
p_panel = "%"
end if
if (qu="") then
qu = "%"
end if
%>
<table>

<tr> <tr valign="top">

<td><table>
<tr>
<td bgcolor="#003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
```



```
TITLE
</font></b></font></div></td>

<TD align="LEFT">
<%
strSQL = "select DISTINCT TITLE_NAME from CORP_INFO WHERE " & Session("PID_PHRASE")
Response.Write "Pid phrase " & Session("PID_PHRASE")
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_TITLE",6, "SELECT TITLE", p_title
objRST.close

%>
</td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
SOURCE CATEGORY
</font></b></font></div></td>
<TD align="LEFT">

<%
strSQL = "select DISTINCT SRC_CAT_NAME from SRC_CATEGORY"
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SCR_CAT",6, "SELECT SRC CAT", p_src_cat
objRST.close
%>

</td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
SOURCE
</font></b></font></div></td>
<TD align="LEFT">
<%
strSQL = "select DISTINCT SOURCE_NAME from SOURCE"
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SOURCE",6, "SELECT SOURCE", p_Source
objRST.close
%>
</td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
CAMPAIGN
</font></b></font></div></td>
<TD align="LEFT"><input type="TEXT" name="P_CAMP" value = '<% Response.Write p_camp %>' ></td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
PANEL
</font></b></font></div></td>
<TD align="LEFT"><input type="TEXT" name="P_PANEL" value = '<% Response.Write p_panel %>' ></td>
</tr>
</table>
</td>
```



```
</tr></table>
<br>
<font face="Verdana, Arial, Helvetica, sans-serif" size="-4">
Break on</font>
<br>
<select name="P_SEL">

<%
dim var(6)

var(1)="SOURCE NAME"
var(2)="CAMPAIGN"
var(3)="PANEL"
for i=1 to 3
  if (cint(p_sel)=i) then %>
    <option value=<%
      Response.Write i %>' selected>
    <%
      Response.Write var(i)
    %>
  </option>
  <%
  else %>
    <option value=<%
      Response.Write i %>>
    <%
      Response.Write var(i)
    %>
  </option>
  <%
  end if
next
%>
</select>
<p>

<input type='IMAGE' Value="UPDATE" Name="UPDATE"
src="..graphics/submit1222.gif" width="54" height="16" border="0">

</p>

</form>
<%
if p_sel > "" then
  SearchResult p_panel, p_camp, p_source, p_src_cat, p_title, p_sel
end if
'Response.Write "query = " & p_sel & " "
'Response.Write "TITLE = " & p_title & " "
'Response.Write "SRC CAT = " & p_src_cat & " "
'Response.Write "TITLE = " & p_source & " "
'Response.Write "<BR>CAMP= " & p_camp & " "
'Response.Write "PANEL = " & p_panel & " "

' if sel="SOURCE CATEGORY" then
  'SrcCatSearchResult p_src_cat, p_title, p_grp
'elseif sel="SOURCE NAME" then
  'SourceSearchResult p_source, p_src_cat, p_title, p_grp
'elseif sel="CAMPAIGN" then
  'CampSearchResult p_camp, p_source, p_src_cat, p_title, p_grp
'elseif sel="PANEL" then
  'PanelSearchResult p_panel, p_camp, p_source, p_src_cat, p_title, p_grp

%>

</BODY>
</HTML>
```





adminsearch2.asp

```
<%@ Language=VBScript %>

    <!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE> IMT Admin SEARCH 2</title>
</HEAD>
<BODY>

<form method="post" action="" id=form1 name=form1>

<% ' our local variables
    sel      = Request.Form("SEL")
    p_grp    = Request.Form("p_grp")
    p_title  = Request.Form("p_title")
    p_src_cat = Request.Form("p_src_cat")
    p_source = Request.Form("p_source")
    p_camp   = Request.Form("p_camp")
    p_Panel  = Request.Form("p_panel")
    if p_grp < " " then
        p_grp = "%"
    end if
    if p_title < " " then
        p_title = "%"
    end if
    if p_src_cat < " " then
        p_src_cat = "%"
    end if
    if p_Source < " " then
        p_Source = "%"
    end if
    if p_camp < " " then
        p_camp = "%"
    end if
    if p_panel < " " then
        p_panel = "%"
    end if
    if (qu="") then
        qu = "%"
    end if
%>
<table>
<tr><td>Product Group</td>
<td><input type="TEXT" name="p_grp" value = '<% Response.Write p_grp %>' ></td>
</tr>
<tr><td>Title Name</td>
<td><input type="TEXT" name="p_title" value = '<% Response.Write p_title %>' ></td>
</tr>
<tr><td>Source Category </td>
<td><input type="TEXT" name="p_src_cat" value = '<% Response.Write p_src_cat %>' ></td>
</tr>
<tr><td>Source </td>
<td><input type="TEXT" name="p_source" value = '<% Response.Write p_source %>' ></td>
</tr>
<tr><td>Campaign </td>
<td><input type="TEXT" name="p_camp" value = '<% Response.Write p_camp %>' ></td>
</tr>
<tr><td>Panel </td>
<td><input type="TEXT" name="p_panel" value = '<% Response.Write p_panel %>' ></td>
</tr>
```



```
</table>
<br>Break on<br>
<select name="SEL">

<%
dim var(6)

var(0) = "PRODUCT GROUP"
var(1) = "TITLE NAME"
var(2) = "SOURCE CATEGORY"
var(3) = "SOURCE NAME"
var(4) = "CAMPAIGN"
var(5) = "PANEL"
for i=0 to 5
    if (sel=var(i)) then
        Response.Write "<option selected>" & var(i) & "</option>"
    else
        Response.Write "<option>" & var(i) & "</option>"
    end if
next
%>
</select>
<br>
<input type="submit" id=submit1 name=submit1>
<br>

</form>
<%

Response.Write "qu = " & p_grp & ", "
Response.Write p_title & ", "
Response.Write p_src_cat & ", "
Response.Write p_source & ", "
Response.Write p_camp & ", "
Response.Write p_panel & " "
Response.Write ", sel= " & Request.Form("SEL")
    if sel = "PRODUCT GROUP" then
        GrpSearchResult p_grp
    elseif sel = "TITLE NAME" then
        ProductSearchResult p_title, p_grp
    elseif sel = "SOURCE CATEGORY" then
        SrcCatSearchResult p_src_cat, p_title, p_grp
    elseif sel = "SOURCE NAME" then
        SourceSearchResult p_source, p_src_cat, p_title, p_grp
    elseif sel = "CAMPAIGN" then
        CampSearchResult p_camp, p_source, p_src_cat, p_title, p_grp
    elseif sel = "PANEL" then
        PanelSearchResult p_panel, p_camp, p_source, p_src_cat, p_title, p_grp
    end if
%>

</BODY>
</HTML>
```

allsetuptabs.asp

```
<%@ Language=VBScript %>
<!-- #include file="./support/global_constants.asp" -->
<!-- #include file="./support/dbConnection.asp" -->
<!-- #include file="./support/adoUtils.asp" -->
<!-- #include file="./support/srcTab.asp" -->
<!-- #include file="./support/navigation.asp" -->

<% JSImageFunctions %>

<%
l_this_file = "allSetuptabs.asp"
Session("refer_page") = "allSetupTabs.asp"
Session("refer_Title") = "Set up Table Selection Screen"

if (Request.Form("setupTab") <> "0"
    and Request.Form("setupTab") > " ") then
    Response.Redirect(Request.Form("setupTab"))
end if

mystrSQL = "select access_level from gateway.dbo.usr_access where login = " & Session("U") & "" & " and product_id =
0" & Session("Product_id")
objRST.open mystrSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
if not objRST.eof then
    Session("USR_ACCESS") = objRST(0)
else
    Session("USR_ACCESS") = "None"
end if
objRST.close

checkSessionVars(2)
select case Session("USR_ACCESS")
case "User"
    CanUseAdmin = "false"
case "Manager"
    CanUseAdmin = "True"
case "Supervisor"
    CanUseAdmin = "True"
case else
    CanUseAdmin = "false"
end select
if CanUseAdmin = "false" then Response.Redirect "NoSetupRights.asp"

%>
<HTML>
<HEAD>

<% DisplayIMTFrameset %>

<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">

</HEAD>

<BODY marginwidth=0 leftmargin=0 marginheight=0 topmargin=0 bgcolor="#99CCFF" vlink="#003366" link="#003366">
<form method="post" action="<%Response.Write l_this_file %>" id="form2" name="form2">
    <% setupTableListBox l_this_file %>

</form>

<P>&nbsp;</P>

</BODY>
</HTML>
```

avail.asp

```
<%@ Language=VBScript %>

<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/keys.asp" -->
<!-- #include file="../support/navigation.asp" -->

<% JSImageFunctions %>

<%
    l_this_file = "avail.asp"

    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if
%>

<html>
<head>

<title>IMT Network</title>

</head>

<%

' l_AvailCount = Request.Form("AVAILCOUNT")
' Response.Write "availCount = " & l_availcount & "<br>"
'on error resume next

dim l_query
dim l_pos(10)
dim l_display
l_display = Request.Form("DISPLAY")
l_query = " select full_promo_key MY_KEY, REC_ID " _
    & " from avail_keys " _
    & " where Product_id = " & session("PRODUCT_ID")

l_query = " select full_promo_key MY_KEY, REC_ID " _
    & " from avail_keys " _
    & " where Product_id = " & session("PRODUCT_ID")
    & " and source_identifier = " & session("SRC_INDICATOR") & " " _
    & " Order by "

for i = 1 to 10
    l_pos(i) = Request.form("KeyPosition" & i)
    if l_pos(i) > " " then
        'SQL
        l_query = l_query & " and POSITION" & i _
            & " = " & Request.Form("KeyPosition" & i) & " "
        'Order by clause
        runMe = true
    end if
    OrderBy = OrderBy & " Position" & i & ", "
next

if (Request.Form("runMe") = "1" ) then
    runMe = true
end if
%>
```



```
end if

Orderby = ORderby & " Full_Promo_key "

    'if (l_pos(1) < " ") then
        'l_pos(1) = Session("SRC_INDICATOR")
        'l_query = l_query & " and POSITION1" _
        ' & " " & l_pos(1) & " "
    end if

    l_query = l_query & " " & OrderBy

%>

<% if Request("add")="1" then %>

<body bgcolor="#99CCFF" vlink="000000" link="003366"
marginwidth="0" marginheight="0" leftmargin="0" topmargin="0"
onLoad="refresh();">

<% DisplayIMTFrameset %>

<% else %>

<body bgcolor="#99CCFF" vlink="000000" link="003366"
marginwidth="0" marginheight="0" leftmargin="0" topmargin="0" >

<% DisplayIMTFrameset %>

<%
end if
%>

<table width="185" cellpadding="2">
<tr>
<td colspan="3" height="39" width="152">
<% keySearchBoxes l_pos,l_this_file, false %>
<tr valign="top" align="right">
<td colspan="4">
<div align="left"><br>
</div>
<div align="left"></div>
<p>
</form>
<form method="post" action="<% Response.Write L_THIS_FILE %>" name="myForm" id=form1>

<% for i = 1 to 10 %>
<input type="HIDDEN"
name="KeyPosition<% Response.Write i %>"
value="<% Response.write l_pos(i) %>"
<% next %>
</p>
</td>
</tr>

<tr>
<td width="162">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<font size="1" color="#000000">
Product: <b><%
Response.write Session("TITLE_NAME") & "-" & Session("PRODUCT_ID")
%> </b>
</font></font> </td>
</tr>
</tr>
```



```
<td width="162">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<font size="1" color="#000000">
<A HREF="/ImtAdmin/admin2a.asp?Fr=<%Response.Write l_this_file %>">
Change Product Id </a>
    </b>
</font></font> </td>
</tr>
<tr valign="top">
<td colspan="4" align="left"> <br>

    <input type="HIDDEN" name="DISPLAY" value="1">
    <%
    'Response.Write l_query & "<BR>"
    if l_display = "1" then

    AvailKeysTable(l_query)
    end if %>

</td>
</tr>
</table>
</form>
</body>
</html>
```



drill.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/drillLib.asp" -->
<!-- #include file="../support/navigation.asp" -->

<% JSImageFunctions %>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY marginwidth=0 leftmargin=0 marginheight=0 topmargin=0 bgcolor="#99CCFF" vlink="#003366" link="#003366">

<% DisplayIMTFrameset %>

<font face="Verdana, Arial, Helvetica, sans-serif" size="-4">
<h3>Fast Find</h3>
<%=Session("PID_PHRASE") %>
<form method="post" action="drill.asp#cur" id="form2" name="form2">
<% printList 1,"x" %>
</form>
</FONT>
</BODY>
</HTML>
```



drillbounce.asp

```

<%@ Language=VBScript %>
<!-- #include file=../support/global_constants.asp -->
<!-- #include file=../support/dbConnection.asp -->
<!-- #include file=../support/adoUtils.asp -->
<!-- #include file=../support/drillLib.asp -->
<%
    l_level = Request.QueryString("Y")
    l_rec_id = Request.QueryString("Z")
    l_option = request.QueryString("O")

    if l_level = 1 then
        mySql = "select grp_name " _
            & " from v_grp_names " _
            & " where min_rec_id = " & l_rec_id
        ObjRST.open mySql, strSQLOLEDB, adOpenStatic, adLockOptimistic
        if not ObjRST.EOF then
            Session("GRP_NAME") = objRST.Fields(0)
        end if
        ObjRST.close

    elseif l_level = 2 then
        mySql = "select grp_name, product_id, title_name " _
            & " from corp_info " _
            & " where rec_id = " & l_rec_id
        ObjRST.open mySql, strSQLOLEDB, adOpenStatic, adLockOptimistic
        if not ObjRST.EOF then
            Session("GRP_NAME") = objRST.Fields(0)
            Session("PRODUCT_ID") = objRST.Fields(1)
            Session("TITLE_NAME") = objRST.Fields(2)
        end if
        ObjRST.close

    elseif l_level = 3 then
        mySql = "select grp_name, product_id, title_name, SRC_CAT_NAME " _
            & " from SRC_CATEGORY " _
            & " where rec_id = " & l_rec_id
        ObjRST.open mySql, strSQLOLEDB, adOpenStatic, adLockOptimistic
        if not ObjRST.EOF then
            Session("GRP_NAME") = objRST.Fields(0)
            Session("PRODUCT_ID") = objRST.Fields(1)
            Session("TITLE_NAME") = objRST.Fields(2)
            Session("SRC_CAT_NAME") = objRST.Fields(3)
        end if
        ObjRST.close

    elseif l_level = 4 then
        mySql = "select grp_name, product_id, title_name, SRC_CAT_NAME, Source_name, SRC_INDICATOR " _
            & " from Source " _
            & " where rec_id = " & l_rec_id
        ObjRST.open mySql, strSQLOLEDB, adOpenStatic, adLockOptimistic
        if not ObjRST.EOF then
            Session("GRP_NAME") = objRST.Fields(0)
            Session("PRODUCT_ID") = objRST.Fields(1)
            Session("TITLE_NAME") = objRST.Fields(2)
            Session("SRC_CAT_NAME") = objRST.Fields(3)
            Session("SOURCE_NAME") = objRST.Fields(4)
            Session("SRC_INDICATOR") = objRST.Fields(5)
        end if
        ObjRST.close

    elseif l_level = 5 then
        mySql = "select grp_name, product_id, title_name, SRC_CAT_NAME, Source_name, SRC_INDICATOR, CAMP_name " _
            & " from Campaigns " _
            & " where rec_id = " & l_rec_id

```

```

ObjRST.open mySql, strSQLOLEDB, adOpenStatic, adLockOptimistic
if not ObjRST.EOF then
    Session("GRP_NAME") = objRST.Fields(0)
    Session("PRODUCT_ID") = objRST.Fields(1)
    Session("TITLE_NAME") = objRST.Fields(2)
    Session("SRC_CAT_NAME") = objRST.Fields(3)
    Session("SOURCE_NAME") = objRST.Fields(4)
    Session("SRC_INDICATOR") = objRST.Fields(5)
    Session("CAMP_NAME") = objRST.Fields(6)
end if
ObjRST.close

elseif l_level = 6 then
    mySql = "select grp_name, product_id, title_name, SRC_CAT_NAME, Source_name, SRC_INDICATOR, CAMP_name,
Panel_name " _
    & "from Panels " _
    & " where rec_id = " & l_rec_id
    ObjRST.open mySql, strSQLOLEDB, adOpenStatic, adLockOptimistic
    if not ObjRST.EOF then
        Session("GRP_NAME") = objRST.Fields(0)
        Session("PRODUCT_ID") = objRST.Fields(1)
        Session("TITLE_NAME") = objRST.Fields(2)
        Session("SRC_CAT_NAME") = objRST.Fields(3)
        Session("SOURCE_NAME") = objRST.Fields(4)
        Session("SRC_INDICATOR") = objRST.Fields(5)
        Session("CAMP_NAME") = objRST.Fields(6)
        Session("PANEL_NAME") = objRST.Fields(7)
    end if
    ObjRST.close

else
    isBad = true
end if

if not isBad then
    if l_option = "EDIT" then
        if l_level = 4 then
            Response.Redirect "editSource.asp?Z=" & l_rec_id
        elseif level = 5 then
            Response.Redirect "editCamp.asp?Z=" & l_rec_id
        elseif l_level = 6 then
            Response.Redirect "editPanel.asp?Z=" & l_rec_id
        end if
    else
        if l_level = 6 then
            Response.Redirect("AdminKeys.htm")
        else
            Response.Redirect("admin" & (l_level + 1) & ".asp")
        end if
    end if
end if

%>

```

edit\_autosetup.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->

<%

If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.form("DELETE.x") <> "" then
    l_button = "Delete"
end if

l_rec_id = Request.QueryString("Z")
Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then
    if l_button = "Update" or l_button = "Delete" then
        l_action = true
        l_src_cat_name = Request.Form("P_SRC_CAT_NAME")
        l_src_ind = Request.Form("P_SRC_IND")
        l_source = Request.Form("P_SOURCE")
        'response.write(l_source)
        objRST.open "select src_cat_name from source where source_name = " & l_source & "", strSQLOLEDB, adOpenStatic,
adLockOptimistic
        if not objRST.eof then l_src_cat_name = objRST(0) else l_src_cat_name = "UNNASSIGNED"
        objRST.close
        if l_src_cat_name = "UNNASSIGNED" then l_source = "UNNASSIGNED"
        'response.write(l_src_cat_name)
        l_valid_from = Request.Form("P_VALID_FROM")
        l_valid_to = Request.Form("P_VALID_TO")
        'response.write (l_valid_to & " < fjdksal;fjdksl;a jfkdl;jas fkd;l;saj fkd fdjksal; fjdksal;fj dk sl;fasjl; ")
        l_setup_mask = Request.Form("P_SETUP_MASK")
        'response.write(l_setup_mask)
        l_e_set_name = Request.Form("P_E_SET_NAME")
        if l_e_set_name = "0" then l_e_set_name = null
        l_login = Session("U")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_autosetup"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_src_cat_name
        objCMD.Parameters(2).Value = l_src_ind
        objCMD.Parameters(3).Value = l_source
        objCMD.Parameters(4).Value = l_valid_from
        objCMD.Parameters(5).value = l_valid_to
        objCMD.Parameters(6).value = l_setup_mask
        objCMD.Parameters(7).value = l_e_set_name
        objCMD.Parameters(8).value = l_login
        objCMD.Parameters(9).value = cint(l_rec_id)
        if l_button = "Delete" then objCMD.Parameters(10).Value = 1 else objCMD.Parameters(10).Value = 0
        Set TempNonSessionVar = objCMD.Execute()
        Response.Redirect "tabautosetup.asp"
        l_action = true
    else
        l_action = false
        strSQL = "select src_cat_name, src_indicator, source_name, valid_from_date, valid_to_date, setup_mask, e_set_name " & _
            " from autosetup_scheme " & _
            " where rec_id = " & l_rec_id
        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If not objRST.EOF then
            l_src_cat_name = ObjRST("src_cat_name")
            l_src_ind = ObjRST("src_indicator")
            l_source = ObjRST("source_name")
            l_valid_from = ObjRST("valid_from_date")

```



```

        l_valid_to = ObjRST("valid_to_date")
        l_setup_mask = ObjRST("setup_mask")
        l_e_set_name = ObjRST("e_set_name")
    end if
    ObjRST.close
end if
end if

%>
<html>
<head>
<title>edit_autosetup.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
<form method="post" action="edit_autosetup.asp?Z=<%Response.Write l_rec_id %>" id=form1 name=form1>
<table width="640" border="0" cellpadding="2">
<% prevSelections(3) %>

<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<table width="500" border="0" cellspacing="3" cellpadding="2">
<tr valign="top">
<td height="25" align="left">
<div align="left"></div>
</td>
<td height="25" align="left">
<input type="IMAGE" Value="UPDATE" Name="UPDATE"
src="..\graphics\submit1222.gif" width="54" height="16" border="0">

<input type="IMAGE" Value="DELETE" Name="DELETE"
src="..\graphics\delete1222.gif" width="53" height="16" border="0">
<%>
<input type="IMAGE" Value="DELETE" Name="DELETE"
'src="..\graphics\delete1222.gif" width="53" height="16" border="0">
<font face="Verdana, Arial, Helvetica, sans-serif" size="4">
<a href="tabautosetup.asp">Return to Auto-Setup table</a></font></td>
</tr>

<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
SOURCE NAME
</font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="000000">
<%if not l_action then
strSQL = "select U_SOURCE_NAME FROM Source_descriptions "
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SOURCE",6, "SELECT SOURCE", l_source
objRST.close
else
Response.Write l_source
end if %>
</font></b></font></div>

</td>
</tr>

```



```
<tr>
  <td width="140" bgcolor="#003366" height="20" align="center">
    <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">SOURCE
    INDICATOR </font></b></font></div>
  </td>
  <td width="360" bgcolor="#99CCFF" height="20" align="left">
    <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

      <%if not l_action then %>
        <input type="TEXT" NAME="P_SRC_IND"
        id=0 value="<%Response.Write l_src_ind %>" maxlength="30" size="6">
      <% else
        Response.Write l_src_ind
      end if %>
    </font></b></font></div>

  </td>
</tr>

  <tr>
    <td width="140" bgcolor="#003366" height="20" align="center">
      <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">VALID
FROM
      </font></b></font></div>
    </td>
    <td width="360" bgcolor="#99CCFF" height="20" align="left">
      <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

        <%if not l_action then %>
          <input type="TEXT" NAME="P_VALID_FROM"
          id=0 value="<%Response.Write l_valid_from %>" maxlength="30"
size="15" width="30">
        <% else
          Response.Write l_valid_from
        end if %>
      </font></b></font></div>

    </td>
  </tr>

  <tr>
    <td width="140" bgcolor="#003366" height="20" align="center">
      <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1">VALID TO
      </font></b></font></div>
    </td>
    <td width="360" bgcolor="#99CCFF" height="20" align="left">
      <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1" color="#000000">

        <%if not l_action then %>
          <input type="TEXT" NAME="P_VALID_TO"
          id=0 value="<%Response.Write l_valid_to %>" maxlength="30" size="15">
        <% else
          Response.Write l_valid_to
        end if %>
      </font></b></font></div>

    </td>
  </tr>

  <tr>
    <td width="140" bgcolor="#003366" height="20" align="center">
      <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1">SETUP MASK
      </font></b></font></div>
    </td>
    <td width="360" bgcolor="#99CCFF" height="20" align="left">
      <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1" color="#000000">

        <%if not l_action then %>
          <input type="TEXT" NAME="P_SETUP_MASK"

```



```
id=0 value="<%Response.Write l_setup_mask %>" maxlength="30"
size="15" >

<% else
    Response.Write l_setup_mask
end if %>
</font></b></font></div>

</td>
</tr>

<tr>
    <td width="140" bgcolor="#003366" height="20" align="center">
        <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF"><b><font size="1">EXPENSE SET
        </font></b></font></div>
    </td>
    <td width="360" bgcolor="#99CCFF" height="20" align="left">
        <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF"><b><font size="1" color="#000000">
            <%if not l_action then
                strSQL = "select e_set_name FROM expense_sets where product_id = " &
Session("PRODUCT_ID") & " order by e_set_name "
                objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
                showQueryListBox objRST, "P_E_SET_NAME",6, "SELECT EXPENSE
SET", l_e_set_name
                objRST.close
            else
                Response.Write l_e_set_name
            end if %>
        </font></b></font></div>
    </td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

edit\_campaign\_def.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->

<%

If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.form("DELETE.x") <> "" then
    l_button = "Delete"
end if

l_rec_id = Request.QueryString("Z")
Response.Write l_rec_id
If (IsNumeric(l_rec_id) ) then
    if l_button = "Update" or l_button = "Delete" then
        l_action = true
        l_campaign_code = Request.Form("P_CAMPAIGN_CODE")
        l_campaign_description = Request.form("P_CAMPAIGN_DESCRIPTION")
        l_src_cat_name = Request.Form("P_SRC_CAT_NAME")
        if not (l_src_cat_name > " ") then l_src_cat_name = "Undefined"
        l_campaign_date = Request.form("P_CAMPAIGN_DATE")
        if not (l_campaign_date > " ") then l_campaign_date = null
        l_panel_scheme = Request.form("P_PANEL_SCHEME")
        if l_panel_scheme = "0" then l_panel_scheme = ""
        l_login = Session("U")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_autosetup_campaign_def"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_campaign_code
        objCMD.Parameters(2).Value = l_campaign_description
        objCMD.Parameters(3).Value = l_src_cat_name
        objCMD.Parameters(4).Value = l_campaign_date
        objCMD.Parameters(5).Value = l_panel_scheme
        objCMD.Parameters(6).value = cint(l_rec_id)
        if l_button = "Delete" then objCMD.Parameters(7) Value = 1 else objCMD.Parameters(7).Value = 0
        Set TempNonSessionVar = objCMD.Execute()
        Response.Redirect "tabautosetup.asp"
        l_action = true
    else
        l_action = false
        strSQL = "select campaign_code, campaign_description, src_cat_name, campaign_date, panel_scheme " & _
            " from autosetup_campaign_def " & _
            " where rec_id = " & l_rec_id
        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_campaign_code = objRST("campaign_code")
            l_campaign_description = objRST("campaign_description")
            l_src_cat_name = objRST("src_cat_name")
            l_campaign_date = objRST("campaign_date")
            l_panel_scheme = objRST("panel_scheme")
        end if
        ObjRST.close
    end if
end if

%>
<html>
<head>
<title>edit_autosetup_campaign_def.asp</title>
```



```
</head>

<body bgcolor="#99CCFF" vlink="#000000" link="#003366">
<form method="post" action="edit_campaign_def.asp?Z=<%Response.Write l_rec_id %>" id=form1 name=form1>
<table width="640" border="0" cellpadding="2">

  <tr valign="top">
    <td align="left" width="640" colspan="3"></td>
  </tr>
  <tr valign="top">
    <td align="left" width="640" colspan="3">
      <table width="500" border="0" cellspacing="3" cellpadding="2">
        <tr valign="top">
          <td height="25" align="left">
            <div align="left"></div>
          </td>
          <td height="25" align="left">
            <input type="IMAGE" Value="UPDATE" Name="UPDATE"
              src="..\graphics/submit1222.gif" width="54" height="16" border="0">
            
            <input type="IMAGE" Value="DELETE" Name="DELETE"
              src="..\graphics/delete1222.gif" width="53" height="16" border="0">
            <%> <input type="IMAGE" Value="DELETE" Name="DELETE"
              'src="..\graphics/delete1222.gif" width="53" height="16" border="0"> %>
            <font face="Verdana, Arial, Helvetica, sans-serif" size="4">
            <a href="tabautosetup.asp">Return to Auto-Setup table</a></font></td>
          </tr>

          <tr>
            <td bgcolor="#003366" height="20" width="140" align="center">
              <div align="center">
                <font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
                <b><font size="1">
                CAMPAIGN CODE
                </font></b></font></div>
              </td>
              <td bgcolor="#99CCFF" height="20" width="360" align="left">
                <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">

                  <%if not l_action then %>
                    <input type="TEXT" NAME="P_CAMPAIGN_CODE"
                      id=0 value="<%Response.Write l_campaign_code %>" maxlength="30"
size="6">

                  <% else
                    Response.Write l_campaign_code
                  end if %>
                  </font></b></font></div>

                </td>
              </tr>

          <tr>
            <td width="140" bgcolor="#003366" height="20" align="center">
              <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">CAMPAIGN
DESCRIPTION</font></b></font></div>
            </td>
            <td width="360" bgcolor="#99CCFF" height="20" align="left">
              <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">

                <%if not l_action then %>
                  <input type="TEXT" NAME="P_CAMPAIGN_DESCRIPTION"
                    id=0 value="<%Response.Write l_campaign_description %>"
maxlength="100" size="100"

                <% else
                  Response.Write l_campaign_description
                end if %>
```





```
</font></b></font></div>
</td>
</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">SOURCE
CATEGORY
</font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<%if not l_action then
<input type="TEXT" NAME="P_SRC_CAT_NAME"
strSQL = "select distinct src_cat_name from source where product_id = " &
Session("PRODUCT_ID") & " order by src_cat_name"
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SRC_CAT_NAME",6, "SELECT SOURCE
CATEGORY", l_src_cat_name
objRST.close
'id=0 value="<%Response.Write l_source_identifier " maxlength="30"
size="6">
else
Response.Write l_src_cat_name
end if %>
</font></b></font></div>
</td>
</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF"><b><font size="1">CAMPAIGN DATE
</font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF"><b><font size="1" color="#000000">
<%if not l_action then %>
<input type="TEXT" NAME="P_CAMPAIGN_DATE"
id=0 value="<%Response.Write l_campaign_date %>" maxlength="30"
size="30">
<% else
Response.Write l_campaign_date
end if %>
</font></b></font></div>
</td>
</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">PANEL
SCHEME
</font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<%if not l_action then
<input type="TEXT" NAME="P_SRC_CAT_NAME"
strSQL = "select distinct panel_scheme from autosetup_panel_def where
product_id = " & Session("PRODUCT_ID") & " order by panel_scheme"
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_PANEL_SCHEME",6, "SELECT PANEL
SCHEME", l_panel_scheme
objRST.close
```



```
'id=0 value="<%Response.Write l_source_identifier " maxlength="30"
size="6">
else
    Response.Write l_panel_scheme
end if %>
</font></b></font></div>
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

edit\_panel\_def.asp

```
<%@ Language=VBScript %>
<!-- #include file=".support/global_constants.asp" -->
<!-- #include file=".support/dbConnection.asp" -->
<!-- #include file=".support/adoUtils.asp" -->
<!-- #include file=".support/srcTab.asp" -->

<%

If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.form("DELETE.x") <> "" then
    l_button = "Delete"
end if

l_rec_id = Request.QueryString("Z")
Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then
    if l_button = "Update" or l_button = "Delete" then
        l_action = true
        l_product_id = Session("PRODUCT_ID")
        l_panel_scheme = request.form("P_PANEL_SCHEME")
        l_panel_codes = request.form("P_PANEL_CODES")
        l_panel_name = Request.form("P_PANEL_NAME")
        l_login = Session("U")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_autosetup_panel_def"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_panel_scheme
        objCMD.Parameters(2).Value = l_panel_codes
        objCMD.Parameters(3).Value = l_panel_name
        objCMD.Parameters(4).value = cint(l_rec_id)
        if l_button = "Delete" then objCMD.Parameters(5).Value = 1 else objCMD.Parameters(5).Value = 0
        Set TempNonSessionVar = objCMD.Execute()
        Response.Redirect "tabautosetup.asp"
        l_action = true
    else
        l_action = false
        strSQL = "select panel_scheme, panel_codes, panel_name " & _
            " from autosetup_panel_def " & _
            " where rec_id = " & l_rec_id
        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_panel_scheme = objRST("PANEL_SCHEME")
            l_panel_codes = objRST("PANEL_CODES")
            l_panel_name = objRST("PANEL_NAME")
        end if
        ObjRST.close
    end if
end if

%>
<html>
<head>
<title>edit_panel_def.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
<form method="post" action="edit_panel_def.asp?Z=<%Response.Write l_rec_id %>" id=form1 name=form1>
<table width="640" border="0" cellpadding="2">
```

```

<tr valign="top">
  <td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
  <td align="left" width="640" colspan="3">
    <table width="500" border="0" cellspacing="3" cellpadding="2">
      <tr valign="top">
        <td height="25" align="left">
          <div align="left"></div>
        </td>
        <td height="25" align="left">
          <input type="IMAGE" Value="UPDATE" Name="UPDATE"
            src="..\graphics/submit1222.gif" width="54" height="16" border="0">
          
          <input type="IMAGE" Value="DELETE" Name="DELETE"
            src="..\graphics/delete1222.gif" width="53" height="16" border="0">
          <%> <input type="IMAGE" Value="DELETE" Name="DELETE"
            'src="..\graphics/delete1222.gif" width="53" height="16" border="0"> %>
          <font face="Verdana, Arial, Helvetica, sans-serif" size="4">
          <a href="tabautosetup.asp">Return to Auto-Setup table</a></font></td>
        </tr>

      <tr>
        <td bgcolor="#003366" height="20" width="140" align="center">
          <div align="center">
            <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
            <b><font size="1">
              PANEL SCHEME
            </font></b></font></div>
          </td>
          <td bgcolor="#99CCFF" height="20" width="360" align="left">
            <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

              <%if not l_action then %>
                <input type="TEXT" NAME="P_PANEL_SCHEME"
                  id=0 value="<%Response.Write l_panel_scheme %>" maxlength="50"
size="30">

              <% else
                Response.Write l_panel_scheme
              end if %>
            </font></b></font></div>

            </td>
          </tr>

      <tr>
        <td width="140" bgcolor="#003366" height="20" align="center">
          <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">PANEL
CODES</font></b></font></div>
        </td>
        <td width="360" bgcolor="#99CCFF" height="20" align="left">
          <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

              <%if not l_action then %>
                <input type="TEXT" NAME="P_PANEL_CODES" id=0
value="<%Response.Write l_panel_codes %>" maxlength="250" size="100">
              <% else
                Response.Write l_panel_codes
              end if %>
            </font></b></font></div>

            </td>
          </tr>

      <tr>
        <td width="140" bgcolor="#003366" height="20" align="center">
          <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">PANEL
NAME

          </font></b></font></div>

```



```
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
  <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
    <%if not l_action then %>
      <input type="TEXT" NAME="P_PANEL_NAME" id=0
value="<%Response.Write l_panel_name %> " maxlength="50" size="30">
    <% else
      Response.Write l_panel_name
    end if %>
  </font></b></font></div>
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

edit\_renewal\_psdef.asp

```

<%@ Language=VBScript %>
<!-- #include file=../support/global_constants.asp -->
<!-- #include file=../support/dbConnection.asp -->
<!-- #include file=../support/adoUtils.asp -->
<!-- #include file=../support/srcTab.asp -->

<%

If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.form("DELETE.x") <> "" then
    l_button = "Delete"
end if

l_rec_id = Request.QueryString("Z")
Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then
    if l_button = "Update" or l_button = "Delete" then
        l_action = true
        l_src_cat_name = Request.Form("P_SRC_CAT_NAME")
        l_source_name = Request.Form("P_SOURCE_NAME")
        if not(l_source_name > " ") then l_source_name = "Undefined"
        l_source_indicator = Request.Form("P_SOURCE_INDICATOR")
        l_prior_source_indicator = Request.form("P_PRIOR_SOURCE_INDICATOR")
        l_valid_from = Request.form("P_VALID_FROM")
        l_valid_to = Request.form("P_VALID_TO")
        objRST.open "select src_cat_name from source where source_name = " & l_source_name & "", strSQLOLEDB,
adOpenStatic, adLockOptimistic
        if not objRST.eof then l_src_cat_name = objRST(0) else l_src_cat_name = "Undefined"
        objRST.close

        response.write (l_prior_source_indicator & " < Prior source indicator ")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_autosetup_renewal_psdef"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_src_cat_name
        objCMD.Parameters(2).Value = l_source_name
        objCMD.Parameters(3).Value = l_source_indicator
        objCMD.Parameters(4).Value = trim(l_prior_source_indicator)
        objCMD.Parameters(5).Value = l_valid_from
        objCMD.Parameters(6).value = l_valid_to
        objCMD.Parameters(7).value = cint(l_rec_id)
        if l_button = "Delete" then objCMD.Parameters(8).Value = 1 else objCMD.Parameters(8).Value = 0
        Set TempNonSessionVar = objCMD.Execute()
        Response.Redirect "tabautosetup.asp"
        l_action = true
    else
        l_action = false
        strSQL = "select source_name, source_indicator, prior_source_indicator, valid_from_date, valid_to_date " & _
            " from autosetup_renewal_psdef " & _
            " where rec_id = " & l_rec_id
        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_source_name = ObjRST("source_name")
            l_source_indicator = objRST("source_indicator")
            l_prior_source_indicator = objRST("prior_source_indicator")
            l_valid_from = ObjRST("valid_from_date")
            l_valid_to = ObjRST("valid_to_date")
        end if
        ObjRST.close
    end if
end if
end if

```

```

%>
<html>
<head>
<title>edit_renewal_psdef.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="#000000" link="#003366">
<form method="post" action="edit_renewal_psdef.asp?Z=<%Response.Write l_rec_id %>" id="form1" name="form1">
<table width="640" border="0" cellpadding="2">
<tr valign="top">
<td align="left" width="640" colspan="3"></td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<table width="500" border="0" cellspacing="3" cellpadding="2">
<tr valign="top">
<td height="25" align="left">
<div align="left"></div>
</td>
<td height="25" align="left">
<input type="IMAGE" Value="UPDATE" Name="UPDATE"
src="..\graphics\submit1222.gif" width="54" height="16" border="0">

<input type="IMAGE" Value="DELETE" Name="DELETE"
src="..\graphics\delete1222.gif" width="53" height="16" border="0">
<%> <input type="IMAGE" Value="DELETE" Name="DELETE"
'src="..\graphics\delete1222.gif" width="53" height="16" border="0"> %>
<font face="Verdana, Arial, Helvetica, sans-serif" size="-4">
<a href="tabautosetup.asp">Return to Auto-Setup table</a></font></td>
</tr>

<tr>
<td bgcolor="#003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
SOURCE NAME
</font></b></font></div>
</td>
<td bgcolor="#99CCFF" height="20" width="360" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
<%if not l_action then
strSQL = "select U_SOURCE_NAME FROM Source_descriptions "
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "P_SOURCE_NAME",6, "SELECT SOURCE",

l_source_name

objRST.close

else
Response.Write l_source_name
end if %>
</font></b></font></div>

</td>

</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">SOURCE
INDICATOR </font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">

<%if not l_action then %>
<input type="TEXT" NAME="P_SOURCE_INDICATOR"

```



```
id=0 value="<%Response.Write l_source_indicator %>" maxlength="30"
size="6">
<% else
    Response.Write l_source_indicator
end if %>
</font></b></font></div>

</td>
</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">PRIOR
SOURCE
INDICATOR </font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

<%if not l_action then %>
    <input type="TEXT" NAME="P_PRIOR_SOURCE_INDICATOR"
    id=0 value="<%Response.Write l_prior_source_indicator %>"
maxlength="250" size="100">
<% else
    Response.Write l_prior_source_indicator
end if %>
</font></b></font></div>

</td>
</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1">VALID
FROM
</font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF"><b><font size="1"
color="#000000">

<%if not l_action then %>
    <input type="TEXT" NAME="P_VALID_FROM"
    id=0 value="<%Response.Write l_valid_from %>" maxlength="30"
size="15">
<% else
    Response.Write l_valid_from
end if %>
</font></b></font></div>

</td>
</tr>

<tr>
<td width="140" bgcolor="#003366" height="20" align="center">
<div align="center"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1">VALID TO
</font></b></font></div>
</td>
<td width="360" bgcolor="#99CCFF" height="20" align="left">
<div align="left"><font face="Verdana, Arial, Helvetica, sans-serif"
color="FFFFFF"><b><font size="1" color="#000000">

<%if not l_action then %>
    <input type="TEXT" NAME="P_VALID_TO"
    id=0 value="<%Response.Write l_valid_to %>" maxlength="30" size="15">
<% else
    Response.Write l_valid_to
end if %>
</font></b></font></div>

</td>
</tr>
```





```
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

editcamp.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->

<%

'      Response.Write "HEY ! ! !"
If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.form("DELETE.x") <> "" then
    l_button = "Delete"
elseif Request.form("COPY") <> "" then
    l_button = "Copy"
end if
l_rec_id = Request.QueryString("Z")
'Response.Write "iodsha ofipasdh floydhopas " & l_button & l_rec_id
If (IsNumeric(l_rec_id)) then
    if l_button = "Update" or l_button = "Delete" or l_button = "Copy" then

'Response.Write " FDSA FDS AFDSA FDSA FDSA FDSA FDSAFDshdf as;hfidashp
fiodhasopfihsio0apfhfdiospahfdiospafi8doas[p fiodsha ofipasdh floydhopas "
        l_camp = REQUEST("P_CAMP")
        l_camp_type = REQUEST("P_CAMP_TYPE")
        l_camp_status = REQUEST("P_CAMP_STATUS")
        l_camp_date_fr = REQUEST("P_CAMP_DATE_FR")
        l_camp_date_to = REQUEST("P_CAMP_DATE_TO")
        l_agt_code = REQUEST("P_AGT_CODE")
        l_agt_descr = REQUEST("P_AGT_DESCR")
        l_e_set_name = REQUEST("P_E_SET_NAME")
        l_r_set_name = REQUEST("P_R_SET_NAME")
        l_NEWSTAND_Sale = REQUEST("P_NEWSTAND_SALE")
        l_budgeted_gross_pct = REQUEST("p_budgeted_gross_pct")
        l_budgeted_vol = REQUEST("p_budgeted_vol")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_campaign"
'      Response.Write l_budgeted_gross_pct & " " & validateNumber(l_budgeted_gross_pct) &"<br>"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_camp
        objCMD.Parameters(2).Value = ValidatePickList(l_camp_type)
        objCMD.Parameters(3).Value = ValidatePickList(l_camp_status)
        objCMD.Parameters(4).Value = validateDate(l_camp_date_fr)
        objCMD.Parameters(5).Value = null
        objCMD.Parameters(6).Value = null
        objCMD.Parameters(7).Value = null
        objCMD.Parameters(8).Value = VALIDATEPickList(l_e_set_name)
        objCMD.Parameters(9).Value = ValidatePickList(l_r_set_name)
        objCMD.Parameters(10).Value = validateNumber(l_NEWSTAND_Sale)
        objCMD.Parameters(11).Value = validateNumber(l_budgeted_gross_pct)
        objCMD.Parameters(12).Value = validateNumber(l_budgeted_vol)
        objCMD.Parameters(13).Value = Session("U")

        objCMD.Parameters(14).Value = l_rec_id
        if l_button = "Delete" then
            ' This is the isDelete Parametere
            objCMD.Parameters(15).Value = 1
        elseif l_button = "Update" then
            objCMD.Parameters(15).Value = 0
        elseif l_button = "Copy" then
            objCMD.Parameters(15).value = 2
        end if

        Set TempNonSessionVar = objCMD.Execute()
    end if
end if

```



```

        Response.Redirect "admin5a.asp?display=1"
    else
        l_action = false
        strSQL = "select CAMP_NAME, CAMP_TYPE, CAMP_STATUS_NAME, " _
        & " CAMP_DATE_FR, CAMP_DATE_TO, AGT_CODE, " _
        & " AGT_DESCR, E_SET_NAME, R_SET_NAME, NEWSTAND_SALE, " _
        & " budgeted_gross_pct, budgeted_vol " _
        & " FROM Campaigns " _
        & " WHERE rec_id = " & cint(l_rec_id)

        ObjRST.open strSQL, strSQLOEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_camp = ObjRST.Fields("CAMP_NAME")
            l_camp_type = ObjRST.Fields("CAMP_TYPE")
            l_camp_status = ObjRST.Fields("CAMP_STATUS_NAME")
            l_camp_date_fr = ObjRST.Fields("CAMP_DATE_FR")
            l_camp_date_to = ObjRST.Fields("CAMP_DATE_TO")
            l_agt_code = ObjRST.Fields("AGT_CODE")
            l_agt_descr = ObjRST.Fields("AGT_DESCR")
            l_e_set_name = ObjRST.Fields("E_SET_NAME")
            l_r_set_name = ObjRST.Fields("R_SET_NAME")
            l_NEWSTAND_Sale = ObjRST.Fields("NEWSTAND_SALE")
            l_budgeted_gross_pct = ObjRST.Fields("budgeted_gross_pct")
            l_budgeted_vol = ObjRST.Fields("budgeted_vol")

        end if
        objRST.close
    end if
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>
<body bgcolor="#99CCFF" vlink="000000" link="003366">

<form method="post" action="editCamp.asp?Z=<%Response.Write l_rec_id %>" id=form1 name=form1>
<table width="640" border="0" cellpadding="2">
<% prevSelections(4) %>
<tr valign="top">
<td align="left" width="640" colspan="3">
</td>
</tr>
<tr valign="top">
<td align="left" width="640" colspan="3">
<table width="500" border="0" cellspacing="3" cellpadding="2">
<tr valign="top">
<td height="25" align="left">
<div align="left"></div>
</td>
<td height="25" align="left">
<input type="IMAGE" Value="UPDATE" Name="UPDATE"
src="./graphics/submit1222.gif" width="54" height="16" border="0">

<SCRIPT LANGUAGE="JavaScript1.1">
<!-- Comment out script for old browsers
function confirm_delete_camp() {
return confirm("This will delete the Campaign, all of its panels and all of its keys. This operation
can not be undone. Are you sure you wish to delete this Campaign?");
}
-->
</SCRIPT>
```



```
<input type='IMAGE' Value="DELETE" Name="DELETE"
src="..\graphics/delete1222.gif" width="53" height="16" onClick="return confirm_delete_camp()" border=0>
</td>
<td><input type="SUBMIT" Value="COPY" Name="COPY" id="COPY">
</td>
<font face="Verdana, Arial, Helvetica, sans-serif" size="4">
<a href="admin5a.asp">Return
to Campaign</a></font></td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
CAMPAIGN NAME
</font></b></font></div>
</td>
<td align="LEFT">
<%if not l_action then %>
<input type="TEXT" NAME="P_CAMP"
id=0 value="<%Response.Write l_camp %>">
<% else
Response.Write l_camp
end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
CAMPAIGN START DATE
</font></b></font></div>
</td>
<td align="LEFT">
<%if not l_action then %>
<input type="TEXT" NAME="P_CAMP_DATE_FR"
id=0 value="<%Response.Write l_camp_date_fr %>">
<% else
Response.Write l_camp_date_fr
end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
CAMPAIGN TYPE
</font></b></font></div>
</td>
<td align="LEFT">
<%if not l_action then
strSQL = "select CAMP_TYPE_NAME from PROD_CAMP_TYPE where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID")
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "p_camp_type",6, "SELECT TYPE", l_camp_type
objRST.close
else
Response.Write l_camp_type
end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
CAMPAIGN STATUS
```



```
</font></b></font></div>
</td>
    <TD align="LEFT">
        <%if not l_action then
            strSQL = "select CAMP_STATUS_NAME from PROD_CAMP_STATUS where " _
            & " PRODUCT_ID=" & Session("PRODUCT_ID")
            objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
            showQueryListBox objRST, "p_camp_status",6, "SELECT STATUS", l_camp_status
            objRST.close
        else
            Response.Write l_camp_status
        end if %>
    </td>
</tr><tr>

</tr>
<% IF SESSION("SRC_CATEGORY") = "Agents" then %>
<tr>
    <td bgcolor="003366" height="20" width="140" align="center">
        <div align="center">
            <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
                <b><font size="1">
                    Agent Code
                </font></b></font></div>
            </td>
            <TD align="LEFT">
                <%if not l_action then %>
                <input type="TEXT" NAME="P_AGT_CODE"
                    id=0 value='<%Response.Write l_agt_code %>'>
                <% else
                    Response.Write l_agt_code
                end if %>
            </td>
        </tr><tr>
            <td bgcolor="003366" height="20" width="140" align="center">
                <div align="center">
                    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
                        <b><font size="1">
                            Agent Descr
                        </font></b></font></div>
                    </td>
                    <TD align="LEFT">
                        <%if not l_action then %>
                        <input type="TEXT" NAME="P_AGT_DESCR"
                            id=0 value='<%Response.Write l_agt_descr%>'>
                        <% else
                            Response.Write l_agt_descr
                        end if %>
                    </td>
                </tr>
            <% else %>

            <input type="hidden" NAME="P_AGT_DESCR"
                value="">
            <input type="hidden" NAME="P_AGT_CODE"
                value="">
            <% end if %>

        <tr>
            <td bgcolor="003366" height="20" width="140" align="center">
                <div align="center">
                    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
                        <b><font size="1">
                            EXPENSE SET
                        </font></b></font></div>
                    </td>
                    <TD align="LEFT">
                        <%if not l_action then
```

```

        strSQL = "select e_set_name from EXPENSE_SETS where " _
        & " PRODUCT_ID=" & Session("PRODUCT_ID")
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "p_e_SET_NAME",6, "SELECT EXPENSE SET", l_e_SET_NAME
objRST.close
else
    Response.Write l_e_set_name
end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
REVENUE SET
</font></b></font></div>
</td>
<td align="LEFT">
<%if not l_action then
    strSQL = "select r_set_name from REVENUE_SETS where " _
    & " PRODUCT_ID=" & Session("PRODUCT_ID")
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "p_r_SET_NAME",6, "SELECT REVENUE SET      ", l_R_SET_NAME
objRST.close
else
    Response.Write l_R_SET_NAME
end if %>

</td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
NEWSSTAND SALE
</font></b></font></div>
</td>
<td align="LEFT">
<%if not l_action then %>
<input type="TEXT" NAME="P_NEWSTAND_SALE"
id=0 value="<%Response.Write l_NEWSTAND_Sale %>">
<% else
    Response.Write l_NEWSTAND_SALE
end if %>

</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
BUDGETED GROSS %
</font></b></font></div>
</td>
<td align="LEFT">
<%if not l_action then %>
<input type="TEXT" NAME="P_budgeted_gross_pct"
id=0 value="<%=l_budgeted_gross_pct%>">
<% else
    Response.Write l_budgeted_gross_pct
end if %>

</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
BUDGETED VOL

```



```
</font></b></font></div>
</td>
  <TD align="LEFT">
    <%if not l_action then %>
      <input type="TEXT" NAME="P_budgeted_vol"
        id=0 value='<%=l_budgeted_vol%>'>
    <% else
      Response.Write l_budgeted_vol
    end if %>
  </td>
</tr>

</table>
</td>
</tr>

</table>
</form>

<P>&nbsp;</P>

</BODY>
</HTML>
```

editcampstat.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%

If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"

elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"

end if

l_ColumnCount = 2
l_rec_id = Request.QueryString("Z")

l_sp_edit = "sp_Edit_Prod_Camp_status"
l_file = "editCampStat.asp"

dim l_col_names(2)
dim l_col_values(2)
l_Title = " Campaign Status"

l_setupTabLink = "setupCampStat.asp"
l_showPrevSelection= false

l_col_names(0) = "CAMPAIGN STATUS"
l_col_names(1) = "DESCRIPTION"

l_query = "select CAMP_STATUS_NAME, CAMP_STATUS_DESCR " _
        & " FROM PROD_camp_status " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then

    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if
end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
```





' call setupTab's editBottom to close body, html, form  
editBottom %>

editcamptvpe.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%
dim l_col_names(2)
dim l_col_values(2)

l_ColumnCount = 2
l_rec_id = Request.QueryString("Z")
If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"

elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"

end if
l_sp_edit = "sp_Edit_Prod_Camp_Type"
l_file = "editCampType.asp"
l_Title = " Campaign Type"
l_setupTabLink = "setupCampType.asp"
l_showPrevSelection= false

l_col_names(0) = "CAMPAIGN TYPE NAME"
l_col_names(1) = "DESCRIPTION"

l_query = "select CAMP_TYPE_NAME, CAMP_TYPE_DESCR " _
        & " FROM PROD_camp_type " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then

    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if

end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>

```

editeset.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%

If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.Form("DELETE.x") <> "" then
    l_button = "Delete"
end if

rec_id = Request.QueryString("Z")
'Response.Write l_rec_id

If IsNumeric(rec_id) then
    'response.write("Hello World > button - " & l_button)
    if l_button = "Update" or l_button = "Delete" then

        product_id = Session("PRODUCT_ID")
        e_set_name = Request.form("e_set_name")
        print_cost = Request.form("print_cost")
        print_cpm_cpu = Request.form("print_cpm_cpu")
        print_basis = Request.form("print_basis")
        merge_purge = Request.form("merge_purge")
        merge_cpm_cpu = Request.form("merge_cpm_cpu")
        merge_basis = Request.form("merge_basis")
        ltr_cost = Request.form("ltr_cost")
        ltr_cpm_cpu = Request.form("ltr_cpm_cpu")
        ltr_basis = Request.form("ltr_basis")
        postage_out_cost = request.form("postage_out_cost")
        postage_out_cpm_cpu = Request.form("postage_out_cpm_cpu")
        postage_out_basis = Request.form("postage_out_basis")
        postage_in_cost = Request.form("postage_in_cost")
        postage_in_cpm_cpu = Request.form("postage_in_cpm_cpu")
        postage_in_basis = Request.form("postage_in_basis")
        premium_cost = Request.form("premium_cost")
        premium_cpm_cpu = Request.form("premium_cpm_cpu")
        premium_basis = Request.form("premium_basis")
        badpay_cost = Request.form("badpay_cost")
        badpay_cpm_cpu = Request.form("badpay_cpm_cpu")
        badpay_basis = Request.form("badpay_basis")
        billing_cost = Request.form("billing_cost")
        billing_cpm_cpu = Request.form("billing_cpm_cpu")
        billing_basis = Request.form("billing_basis")
        subs_svc_costs = Request.form("subs_svc_costs")
        subs_svc_cpm_cpu = Request.form("subs_svc_cpm_cpu")
        subs_svc_basis = Request.form("subs_svc_basis")
        exp_other_costs = Request.form("exp_other_costs")
        exp_other_cpm_cpu = Request.form("exp_other_cpm_cpu")
        exp_other_basis = Request.form("exp_other_basis")
        if l_button = "Delete" then IsDelete = 1 else IsDelete = 0

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_prod_e_set"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters("@product_id").Value = product_id
        objCMD.Parameters("@e_set_name").Value = e_set_name
        if not (print_cost > " ") then
            objCMD.Parameters("@print_cost").Value = null
        else
            objCMD.Parameters("@print_cost").Value = print_cost
        end if
    end if
end if

```

```

objCMD.Parameters("@print_cpm_cpu").Value = print_cpm_cpu
objCMD.Parameters("@print_basis").Value = print_basis
if not (merge_purge > " ") then
    objCMD.Parameters("@merge_purge").Value = null
else
    objCMD.Parameters("@merge_purge").Value = merge_purge
end if
objCMD.Parameters("@merge_cpm_cpu").Value = merge_cpm_cpu
objCMD.Parameters("@merge_basis").Value = merge_basis
if not (ltr_cost > " ") then
    objCMD.Parameters("@ltr_cost").Value = null
else
    objCMD.Parameters("@ltr_cost").Value = ltr_cost
end if
objCMD.Parameters("@ltr_cpm_cpu").Value = ltr_cpm_cpu
objCMD.Parameters("@ltr_basis").Value = ltr_basis
if not (postage_out_cost > " ") then
    objCMD.Parameters("@postage_out_cost").value = null
else
    objCMD.Parameters("@postage_out_cost").Value = postage_out_cost
end if
objCMD.Parameters("@postage_out_cpm_cpu").Value = postage_out_cpm_cpu
objCMD.Parameters("@postage_out_basis").Value = postage_out_basis
if not (postage_in_cost > " ") then
    objCMD.Parameters("@postage_in_cost") = null
else
    objCMD.Parameters("@postage_in_cost").Value = postage_in_cost
end if
objCMD.Parameters("@postage_in_cpm_cpu").Value = postage_in_cpm_cpu
objCMD.Parameters("@postage_in_basis").Value = postage_in_basis
if not (premium_cost > " ") then
    objCMD.Parameters("@premium_cost") = null
else
    objCMD.Parameters("@premium_cost").Value = premium_cost
end if
objCMD.Parameters("@premium_cpm_cpu").Value = premium_cpm_cpu
objCMD.Parameters("@premium_basis").Value = premium_basis
if not (badpay_cost > " ") then
    objCMD.Parameters("@badpay_cost").value = null
else
    objCMD.Parameters("@badpay_cost").Value = badpay_cost
end if
objCMD.Parameters("@badpay_cpm_cpu").Value = badpay_cpm_cpu
objCMD.Parameters("@badpay_basis").Value = badpay_basis
if not (billing_cost > " ") then
    objCMD.Parameters("@billing_cost").Value = null
else
    objCMD.Parameters("@billing_cost").Value = billing_cost
end if
objCMD.Parameters("@billing_cpm_cpu").Value = billing_cpm_cpu
objCMD.Parameters("@billing_basis").Value = billing_basis
if not (subs_svc_costs > " ") then
    objCMD.Parameters("@subs_svc_costs").value = null
else
    objCMD.Parameters("@subs_svc_costs").Value = subs_svc_costs
end if
objCMD.Parameters("@subs_svc_cpm_cpu").Value = subs_svc_cpm_cpu
objCMD.Parameters("@subs_svc_basis").Value = subs_svc_basis
if not (exp_other_costs > " ") then
    objCMD.Parameters("@exp_other_costs") = null
else
    objCMD.Parameters("@exp_other_costs").Value = exp_other_costs
end if
objCMD.Parameters("@exp_other_cpm_cpu").Value = exp_other_cpm_cpu
objCMD.Parameters("@exp_other_basis").Value = exp_other_basis
objCMD.Parameters("@rec_id").value = rec_id
objCMD.Parameters("@IsDelete").Value = IsDelete
Set TempNonSessionVar = objCMD.Execute()

```



```
Response.Redirect "setupeset.asp"
else 'Neither Update or Delete was pressed

strSQL = "select * " & _
        " from expense_sets " & _
        " where rec_id = " & rec_id

'response.write (strSQL)
ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
end if
end if

if l_button <> "Update" and l_button <> "Delete" then
%>
<html>
<head>
<title>editset.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
<form method="post" action="editset.asp?Z=<%Response.Write rec_id %>" id=form1 name=form1>
    <table border="0" cellpadding="2">
        <tr>
            <td>
                <input type='IMAGE' Value="UPDATE" Name="UPDATE"
src="..\graphics/submit1222.gif" width="54" height="16" border="0">
                
                <input type='IMAGE' Value="DELETE" Name="DELETE"
src="..\graphics/delete1222.gif" width="53" height="16" border=0>
            </td>
            <td>
                <a href="setupeset.asp">Return to Expense Set Setup Table</a>
            </td>
        </tr>
        <tr>
            <td><td>&nbsp;</td></tr>
    </table>
    <table>
        <tr>
            <td>
                Expense Set
                <input type="TEXT" NAME="e_set_name" value="<%=objRST("e_set_name")%>">
            </td>
            <td>
                Print Cost
                <input type="TEXT" NAME="print_cost" value="<%=objRST("print_cost")%>">
                <% showLOVListBox "CPM,CPU","print_cpm_cpu",2,"", objRST("print_cpm_cpu")
%>
            </td>
            <td>
                <input type="HIDDEN" NAME="print_basis" value="Mail Qty">
                Mail Qty
            </td>
        </tr>
        <tr>
            <td>
                Merge Purge Cost
                <input type="TEXT" NAME="merge_purge" value="<%=objRST("merge_purge")%>">
            </td>
            <td>
            </td>
        </tr>
    </table>
</form>
</body>
</html>
```



```
</td>
<td>
    <% showLOVListBox "CPM,CPU","merge_cpm_cpu",2,"", objRST("print_cpm_cpu")
%>
</td>
<td>
    <% showLOVListBox "Mail Qty,Merge Purge Qty","merge_basis",2,"",
objRST("merge_basis") %>
</td>
</tr>
<tr>
<td>
    Lettershop Cost
</td>
<td>
    <input type="TEXT" NAME="ltr_cost" value="<%=objRST("ltr_cost")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","ltr_cpm_cpu",2,"", objRST("ltr_cpm_cpu") %>
</td>
<td>
    <input type="HIDDEN" NAME="ltr_basis" value="Mail Qty">
    Mail Qty
</td>
</tr>
<tr>
<td>
    Postage Out Cost
</td>
<td>
    <input type="TEXT" NAME="postage_out_cost"
value="<%=objRST("postage_out_cost")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","postage_out_cpm_cpu",2,"",
objRST("postage_out_cpm_cpu") %>
</td>
<td>
    <input type="HIDDEN" NAME="postage_out_basis" value="Mail Qty">
    Mail Qty
</td>
</tr>
<tr>
<td>
    Postage In Cost
</td>
<td>
    <input type="TEXT" NAME="postage_in_cost"
value="<%=objRST("postage_in_cost")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","postage_in_cpm_cpu",2,"",
objRST("postage_in_cpm_cpu") %>
</td>
<td>
    <input type="HIDDEN" NAME="postage_in_basis" value="Gross Subs">
    Gross Subs
</td>
</tr>
<tr>
<td>
    Premium Cost
</td>
<td>
    <input type="TEXT" NAME="premium_cost"
value="<%=objRST("premium_cost")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","premium_cpm_cpu",2,"",
objRST("postage_in_cpm_cpu") %>
```



```

</td>
<td>
    <% showLOVListBox "Gross Subs,Net Subs","premium_basis",2,"",
objRST("premium_basis") %>
</td>
</tr>
<tr>
<td>
    Bad Pay Cost
</td>
<td>
    <input type="TEXT" NAME="badpay_cost" value="<%=objRST("badpay_cost")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","badpay_cpm_cpu",2,"",
objRST("badpay_cpm_cpu") %>
</td>
<td>
    <input type="HIDDEN" NAME="badpay_basis" value="Gross-Net Subs">
    Gross-Net Subs
</td>
</tr>
<tr>
<td>
    Billing Cost
</td>
<td>
    <input type="TEXT" NAME="billing_cost" value="<%=objRST("billing_cost")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","billing_cpm_cpu",2,"", objRST("billing_cpm_cpu")
%>
</td>
<td>
    <input type="HIDDEN" NAME="billing_basis" value="Credit Subs">
    Credit Subs
</td>
</tr>
<tr>
<td>
    Sub Service Cost
</td>
<td>
    <input type="TEXT" NAME="subs_svc_costs"
value="<%=objRST("subs_svc_costs")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","subs_svc_cpm_cpu",2,"",
objRST("subs_svc_cpm_cpu") %>
</td>
<td>
    <input type="HIDDEN" NAME="subs_svc_basis" value="Net Subs">
    Net Subs
</td>
</tr>
<tr>
<td>
    Exp Other Cost
</td>
<td>
    <input type="TEXT" NAME="exp_other_costs"
value="<%=objRST("exp_other_costs")%>">
</td>
<td>
    <% showLOVListBox "CPM,CPU","exp_other_cpm_cpu",2,"",
objRST("exp_other_cpm_cpu") %>
</td>
<td>

```



```

                                <% showLOVListBox "Mail Qty,Merge Purge,Gross Subs,Net
Subs","exp_other_basis",2,"", objRST("exp_other_basis") %>
                                </td>
                                </tr>
                                </table>
                                </td>
                                </tr>
                                </table>
                                <%=l_button%>
                                </form>
                                </body>
                                </html>

                                <% end if %>

```



editkeys.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%
l_debug = 0 ' set to 1 for debugging . . .
l_rec_id = Request.QueryString("Z")
if l_debug = 1 then
    Response.write l_rec_id
end if

If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"
elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"
end if
l_sp_edit = "sp_Edit_Keys"
l_file = "editKeys.asp"
l_ColumnCount = 16
dim l_col_names(16)
dim l_col_values(16)
dim l_col_types(16)
dim l_col_desc1(16)
dim l_col_desc2(16)
l_Title = "Keys "

l_setupTabLink = "adminKeysRight.asp"
l_showPrevSelection= false
l_col_names(0) = "KEY"

l_col_names(1) = "KEY DESCRIPTION"
l_col_names(2) = "MAIL QTY"
l_col_names(3) = "MAIL QTY OVERRIDE"
l_col_names(4) = "LIST CATEGORY"
l_col_names(5) = "LIST NAME"
l_col_names(6) = "LIST SEGMENT"
l_col_names(7) = "LIST CPM"
l_col_names(8) = "LIST COST BASIS"
l_col_names(9) = "LIST COST FACTOR"
l_col_names(10) = "MERGE PURGE QTY"
l_col_names(11) = "EXPENSE SET"
l_col_names(12) = "REVENUE SET"
l_col_names(13) = "PACKAGE"
l_col_names(14) = "SUBTOTAL FLAG"
l_col_names(15) = "OPTIONAL EFFORT ID"

l_query = "select "
l_query = l_query & " FULL_PROMO_KEY , '0
l_query = l_query & " P_KEY_DESCR , '1
l_query = l_query & " key_mail_QTY , '2
l_query = l_query & " MAIL_QTY_OVERRIDE , '3
l_query = l_query & " LIST_CAT_NAME , '4
l_query = l_query & " LIST_NAME , '5 - CHANGE TO LIST NAME
l_query = l_query & " LIST_SEGMENT_NAME , '6 - CHANGE TO LIST SEGMENT
l_query = l_query & " list_cost_actual_cpm , '7 - CHANGE TO LIST SEGMENT
l_query = l_query & " list_cost_basis , '8
l_query = l_query & " list_cost_factor , '9
l_query = l_query & " merge_purge_qty , '10
l_query = l_query & " e_set_name , '11
l_query = l_query & " r_set_name , '12
l_query = l_query & " PANEL_PKG , '13
l_query = l_query & " SUBTOTAL_FLAG , '14
l_query = l_query & " OPTIONAL_EFFORT_ID, FULL_PROMO_KEY, REC_ID " _
& " from Keys " _

```

```

    & " where rec_id = " & clng(l_rec_id)
if l_debug = 1 then
    for m = 1 to 15
        Response.write "P_" & m & " " & request("P_" & m) & "<br>"
    next

Response.write "<BR>" & l_rec_id & "<BR>" & l_query & "<BR>"
end if
If (IsNumeric(l_rec_id)) then

if l_action = "Update" or l_action = "Delete" then
    'editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    ' must skip 0 and 2
    l_0 = Request("P_1")
    l_1 = Request("P_3")
    l_2 = Request("P_4")
    l_3 = Request("P_5")
    l_4 = Request("P_6")
    l_5 = Request("P_7")
    l_6 = Request("P_8")
    l_7 = Request("P_9")
    l_8 = Request("P_10")
    l_9 = Request("P_11")
    l_10 = Request("P_12")
    l_11 = Request("P_13")
    l_12 = Request("P_14")
    l_13 = Request("P_15")

    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_edit_key"

    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh

    '
    USER SUPPLIED
    if l_debug = 1 then
        Response.write "<br>1 " & l_0
        Response.write "<br>2 " & l_1
        Response.write "<br>3 " & validatePickList(l_2)
        Response.write "<br>4 " & validatePickList(l_3)
        Response.write "<br>5 " & validatePickList(l_4)
        Response.write "<br>6 " & validateNumber(l_5)
        Response.write "<br>7 " & validatePickList(l_6)
        Response.write "<br>8 " & validateNumber(l_7)
        Response.write "<br>9 " & validateNumber(l_8)
        Response.write "<br>10 " & validatePickList(l_9)
        Response.write "<br>11 " & validatePickList(l_10)
        Response.write "<br>12 " & validatePickList(l_11)
        Response.write "<br>13 " & validatePickList(l_12)
        Response.write "<br>14 " & l_13
        Response.write "<br>15 " & validatePickList(l_1)
    end if
    SUPPLIED BY ME
    objCMD.Parameters(1).Value = l_0
    objCMD.Parameters(2).Value = VALIDATENUMBER(l_1)
    objCMD.Parameters(3).Value = validatePickList(l_2)
    objCMD.Parameters(4).Value = validatePickList(l_3)
    objCMD.Parameters(5).Value = validatePickList(l_4)
    objCMD.Parameters(6).Value = validateNumber(l_5)
    objCMD.Parameters(7).Value = validatePickList(l_6)
    objCMD.Parameters(8).Value = validateNumber(l_7)
    objCMD.Parameters(9).Value = validateNumber(l_8)
    objCMD.Parameters(10).Value = validatePickList(l_9)
    objCMD.Parameters(11).Value = validatePickList(l_10)
    objCMD.Parameters(12).Value = validatePickList(l_11)
    objCMD.Parameters(13).Value = validatePickList(l_12)
    objCMD.Parameters(14).Value = l_13

```

```

        objCMD.Parameters(15).Value = l_Rec_id
    if l_action = "Delete" then
        ' This is the isDelete Parametere
        objCMD.Parameters(16).Value = 1
    else
        objCMD.Parameters(16).Value = 0
    end if
if l_debug <> 1 then
    Set TempNonSessionVar = objCMD.Execute()

    Response.Redirect "adminkeysright.asp"
end if

else
    strSQL = l_query

    ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    If (ObjRST.EOF = False) then
        l_col_values(0) = ObjRST.Fields(0) ' key
        l_col_values(1) = ObjRST.Fields(1) ' p_KEY_desc
        l_col_values(2) = ObjRST.Fields(2) 'MAIL_QTY
        l_col_values(3) = ObjRST.Fields(3) 'MAIL_QTY_OVERRIDE
        l_col_values(4) = ObjRST.Fields(4) ' LIST_CAT_NAME
        l_col_values(5) = ObjRST.Fields(5) ' LIST_NAME
        l_col_values(6) = ObjRST.Fields(6) ' LIST_SEGMENT_NAME
        l_col_values(7) = ObjRST.Fields(7) ' list_cost_actual_cpm
        l_col_values(8) = ObjRST.Fields(8) ' list_cost_basis
        l_col_values(9) = ObjRST.Fields(9) 'list_cost_factor
        l_col_values(10) = ObjRST.Fields(10) 'merge_purge_qty
        l_col_values(11) = ObjRST.Fields(11) ' e_set_name
        l_col_values(12) = ObjRST.Fields(12) ' r_set_name
        l_col_values(13) = ObjRST.Fields(13) 'PANEL_PKG
        l_col_values(14) = ObjRST.Fields(14) ' subtotal flag
        l_col_values(15) = ObjRST.Fields(15) ' optional_effort_id
        l_key = ObjRST.Fields("FULL_PROMO_KEY")

        if l_debug = 1 then
            Response.Write l_col_values(0)
            Response.Write l_col_values(1) & "<br>"
            Response.Write l_col_values(2) & "<br>"
            Response.Write l_col_values(3) & "<br>"
            Response.Write l_col_values(4) & "<br>"
            Response.Write l_col_values(5) & "<br>"
            Response.Write l_col_values(6) & "<br>"
            Response.Write l_col_values(7) & "<br>"
            Response.Write l_col_values(8) & "<br>"
            Response.Write l_col_values(9) & "<br>"
            Response.Write l_col_values(10) & "<br>"
            Response.Write l_col_values(11) & "<br>"
            Response.Write l_col_values(12) & "<br>"
        end if
    end if
    objRST.close
end if

end if
for i = 0 to 16
    if i = 0 then
        l_col_types(i) = 0
    elseif i = 2 then
        l_col_types(i) = 0
    elseif i = 4 then
        l_col_types(i) = 2
    end if
    l_col_desc1 = "select list_cat_name from list_CATS where "
    & "PRODUCT_ID=" & Session("PRODUCT_ID") & " order by list_cat_name "

```

```

l_col_desc2(i)="SELECT List Cat"
elseif i = 5 then
l_col_types(i) = 2
l_col_desc1(i)="select list_Name from list_names where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID") & " order by list_name "
l_col_desc2(i)="SELECT List Name"
elseif i = 6 then
l_col_types(i) = 2
l_col_desc1(i)="select list_segment_name from list_segments where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID") & " order by list_segment_name "
l_col_desc2(i)="SELECT List Segment"
elseif i = 8 then
l_col_types(i) = 3
l_col_desc1(i)="Mail Qty,Merge Purge Qty"
l_col_desc2(i)="SELECT List Cost Basis"
elseif i = 11 then
l_col_types(i) = 2
l_col_desc1(i)="select e_set_name from EXPENSE_SETS where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID") & " order by e_set_name "
l_col_desc2(i)="SELECT Expense Set"
elseif i = 12 then
l_col_types(i) = 2
l_col_desc1(i)="select r_set_name from REVENUE_SETS where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID") & " order by r_set_name "
l_col_desc2(i)="SELECT Revenue Set"
elseif i = 13 then
l_col_types(i) = 2
l_col_desc1(i)="select package_name from PROD_PANEL_PACKAGE where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID") & " order by package_name "
l_col_desc2(i)="SELECT Panel Pkg"
elseif i = 14 then
l_col_types(i) = 3
l_col_desc1(i)="True,False,Split"
l_col_desc2(i)="SELECT Subtotal Flag"
else
l_col_types(i) = 1
l_col_desc1(i)=""
l_col_desc2(i)=""
end if
next
' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

' call setupTab's editmain to paint screen for user input
editComplex l_Title, l_setupTabLink, " ", l_showPrevSelection, l_col_names, l_col_values, l_col_types, l_col_desc1, l_col_desc2,
l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom

%>

```

editlistcat.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%
l_ColumnCount = 2
l_rec_id = Request.QueryString("Z")
If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"

elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"

end if
l_sp_edit = "sp_Edit_Prod_List_Cat"
l_file = "editListCat.asp"

dim l_col_names(2)
dim l_col_values(2)
l_Title = " List Category "

l_setupTabLink = "setupListCat.asp"
l_showPrevSelection= false

l_col_names(0) = "LIST CATEGORY"
l_col_names(1) = "DESCRIPTION"

l_query = "select LIST_CAT_NAME, LIST_CAT_DESCR " _
        & " FROM List_cats " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then

    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if

end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>
```



editlistname.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%
dim l_col_names(2)
dim l_col_values(2)

l_ColumnCount = 2
l_rec_id = Request.QueryString("Z")
If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"

elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"

end if
l_sp_edit = "sp_Edit_Prod_List_Name"
l_file = "editListName.asp"
l_Title = " List Name "
l_setupTabLink = "setupListName.asp"
l_showPrevSelection= false

l_col_names(0) = "LIST NAME"
l_col_names(1) = "DESCRIPTION"

l_query = "select LIST_NAME, List_Name_description " _
        & " FROM List_names " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then

    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if

end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>
```





editlistsegment.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%
    l_ColumnCount = 2
    l_rec_id = Request.QueryString("Z")
    If Request.Form("UPDATE.x") <> "" then
        l_action = "Update"

    elseif Request.form("DELETE.x") <> "" then
        l_action = "Delete"

    end if
    l_sp_edit = "sp_Edit_Prod_List_Segment"
    l_file = "editListSegment.asp"

    dim l_col_names(2)
    dim l_col_values(2)
    l_Title = " List Segment "

    l_setupTabLink = "setupListSegment.asp"
    l_showPrevSelection= false

    l_col_names(0) = "LIST SEGMENT"
    l_col_names(1) = "DESCRIPTION"

    l_query = "select list_segment_name, list_segment_description " _
        & " FROM List_Segments " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then

    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if

end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>
```



editpanel.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->

<%
l_debug = 0
If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.Form("DELETE.x") <> "" then
    l_button = "Delete"
end if

l_rec_id = Request.QueryString("Z")
Response.Write l_rec_id
If (IsNumeric( l_rec_id )) then
    if l_button = "Update" or l_button = "Delete" then
        l_panel_name = Request("P_PANEL_NAME")
        l_panel_type = Request("P_PANEL_TYPE")
        l_panel_subtype = Request("P_PANEL_SUBTYPE")
        l_test_type = Request("P_TEST_TYPE")
        l_panel_offer = Request("P_PANEL_OFFER")
        l_panel_offer_price = Request("P_PANEL_OFFER_PRICE")
        l_panel_pkg = Request("P_PANEL_PKG")
        l_panel_premium = Request("P_PANEL_PREMIUM")
        l_panel_date_start = Request("P_PANEL_DATE_START")
        l_panel_date_end = Request("P_PANEL_DATE_END")
        l_e_set_name = Request("P_E_SET_NAME")
        l_r_set_name = Request("P_R_SET_NAME")
        l_newstand_sale_marker = Request("P_NEWSTAND_SALE_MARKER")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_panel"
        USER SUPPLIED

        if l_debug = 1 then
            Response.Write "l_panel_name " & l_panel_name & "<BR>"
            Response.Write "l_panel_type " & l_panel_type & "<BR>"
            Response.Write "l_panel_subtype " & l_panel_subtype & "<BR>"
            Response.Write "l_test_type " & l_test_type & "<BR>"
            Response.Write "l_panel_offer " & l_panel_offer & "<BR>"
            Response.Write "l_panel_offer_price " & l_panel_offer_price & "<BR>"
            Response.Write "l_panel_pkg " & l_panel_pkg & "<BR>"
            Response.Write "l_panel_premium " & l_panel_premium & "<BR>"
            Response.Write "l_panel_date_start " & l_panel_date_start & "<BR>"
            Response.Write "l_panel_date_end " & l_panel_date_end & "<BR>"
            Response.Write "l_e_set_name " & l_e_set_name & "<BR>"
            Response.Write "l_r_set_name " & l_r_set_name & "<BR>"
            Response.Write "l_newstand_sale_marker " & l_newstand_sale_marker & "<BR>"
        end if

        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_panel_name
        objCMD.Parameters(2).Value = validateDate( l_panel_date_start )
        objCMD.Parameters(3).Value = ValidatePickList( l_panel_type )
        objCMD.Parameters(4).Value = ValidatePickList( l_panel_subtype )
        objCMD.Parameters(5).Value = ValidatePickList( l_test_type )
        objCMD.Parameters(6).Value = ValidatePickList( l_panel_pkg )
        objCMD.Parameters(7).Value = ValidatePickList( l_e_set_name )
        objCMD.Parameters(8).Value = ValidatePickList( l_r_set_name )
        objCMD.Parameters(9).Value = ValidatePickList( l_newstand_sale_marker )
        objCMD.Parameters(10).Value = session("U")

        SUPPLIED BY ME
        objCMD.Parameters(11).Value = l_rec_id
        if l_button = "Delete" then

```

```

' This is the isDelete Parametere
    objCMD.Parameters(12).Value = 1
else
    objCMD.Parameters(12).Value = 0
end if

Set TempNonSessionVar = objCMD.Execute()
Response.Redirect "admin6a.asp?display=1"

else
    l_action = false
    strSQL = "select PANEL_NAME, PANEL_TYPE, PANEL_SUBTYPE, " _
    & " TEST_TYPE, PANEL_OFFER, PANEL_OFFER_PRICE, PANEL_PKG, " _
    & " PANEL_PREMIUM, PANEL_DATE_START, PANEL_DATE_END, " _
    & " E_SET_NAME, R_SET_NAME, newstand_sale_marker " _
    & " from PANELS " _
    & " where rec_id = " & cint(l_rec_id)

    ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

    If (ObjRST.EOF = False) then
        l_panel_name = objRST.Fields("PANEL_NAME")
        l_panel_type = objRST.Fields("PANEL_TYPE")
        l_panel_subtype = objRST.Fields("PANEL_SUBTYPE")
        l_test_type = objRST.Fields("TEST_TYPE")
        l_panel_offer = objRST.Fields("PANEL_OFFER")
        l_panel_offer_price = objRST.Fields("PANEL_OFFER_PRICE")
        l_panel_pkg = objRST.Fields("PANEL_PKG")
        l_panel_premium = objRST.Fields("PANEL_PREMIUM")
        l_panel_date_start = objRST.Fields("PANEL_DATE_START")
        l_panel_date_end = objRST.Fields("PANEL_DATE_END")
        l_e_set_name = objRST.Fields("E_SET_NAME")
        l_r_set_name = objRST.Fields("R_SET_NAME")
        l_newstand_sale_marker = objRST.Fields("newstand_sale_marker")

    end if
    objRST.close
end if
end if

%>
<html>
<head>
<title>IMT Network</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
<% Response.Write "<BR>" & l_panel_name
' Response.Write "<BR>" & l_panel_type
' Response.Write "<BR>" & l_panel_subtype
' Response.Write "<BR>" & l_test_type
' Response.Write "<BR>" & l_panel_offer
' Response.Write "<BR>" & l_panel_offer_price
' Response.Write "<BR>" & l_panel_pkg
' Response.Write "<BR>" & l_panel_premium
' Response.Write "<BR>" & l_panel_date_start
' Response.Write "<BR>" & l_panel_date_end
' Response.Write "<BR>" & l_e_set_name
' Response.Write "<BR>" & l_r_set_name
%>
<form method="post" action="editPanel.asp?Z=<% Response.Write l_rec_id %>" id=form1 name=form1>
<table width="640" border="0" cellpadding="2">
<% prevSelections(5) %>
<tr valign="top">
<td align="left" width="640" colspan="3">
</td>

```

```

</tr>
<tr valign="top">
  <td align="left" width="640" colspan="3">
    <table width="500" border="0" cellspacing="3" cellpadding="2">
      <tr valign="top">
        <td height="25" align="left">
          <div align="left"></div>
        </td>
        <td height="25" align="left">
          <input type="IMAGE" Value="UPDATE" Name="UPDATE"
            src="..graphics/submit1222.gif" width="54" height="16" border="0">
          
          <SCRIPT LANGUAGE="JavaScript1.1">
            <!-- Comment out script for old browsers
              function confirm_delete_panel() {
                return confirm("This will delete the panel and all its keys. This operation can not be undone. Are you
sure you wish to delete this panel?");
              }
            <!-->
          </SCRIPT>
          <input type="IMAGE" Value="DELETE" Name="DELETE"
            src="..graphics/delete1222.gif" width="53" height="16" onClick="return confirm_delete_panel()" border="0">

          <font face="Verdana, Arial, Helvetica, sans-serif"
            size="4">
            <a href="admin6a.asp">Return
              to Panel</a></font></td>
          </tr>
        <tr>
          <td bgcolor="003366" height="20" width="140" align="center">
            <div align="center">
              <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
                <b><font size="1">
                  PANEL NAME
                </font></b></font></div>
            </td>
            <td align="left">
              <%if not l_action then %>
                <input type="TEXT" NAME="P_PANEL_NAME"
                  id=0 value='<%Response.Write l_panel_name %>' maxlength=250>
              <% else
                Response.Write l_panel_name
              end if %>
            </td>
          </tr><tr>
            <td bgcolor="003366" height="20" width="140" align="center">
              <div align="center">
                <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
                  <b><font size="1">
                    PANEL START DATE
                  </font></b></font></div>
            </td>
            <td align="left">
              <%if not l_action then %>
                <input type="TEXT" NAME="P_PANEL_DATE_START"
                  id=0 value='<%Response.Write l_panel_date_start %>'>
              <% else
                Response.Write l_panel_date_start
              end if %>
            </td>
          </tr><tr>
            <td bgcolor="003366" height="20" width="140" align="center">
              <div align="center">
                <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
                  <b><font size="1">
                    PANEL TYPE
                  </font></b></font></div>
            </td>

```

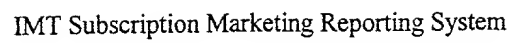


```
<TD align="left">
  <%if not l_action then
    strSQL = "select PANEL_TYPE_NAME from CORP_PANEL_TYPE"
    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    showQueryListBox objRST, "p_panel_type", 6, "SELECT TYPE", l_panel_type
    objRST.close
  else
    Response.Write l_panel_type
  end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
  PANEL SUBTYPE
</font></b></font></div>
</td>
<TD align="left">
  <%if not l_action then
    strSQL = "select PANEL_SUB_TYPE_NAME from CORP_PANEL_SUB_TYPE"
    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    showQueryListBox objRST, "p_panel_subtype", 6, "SELECT SUB TYPE", l_panel_subtype
    objRST.close
  else
    Response.Write l_panel_subtype
  end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
  PANEL TEST TYPE
</font></b></font></div>
</td>
<TD align="left">
  <%if not l_action then
    strSQL = "select PANEL_TEST_TYPE_NAME from CORP_PANEL_TEST_TYPE"
    objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    showQueryListBox objRST, "p_test_type", 6, "SELECT TEST TYPE", l_test_type
    objRST.close
  else
    Response.Write l_test_type
  end if %>
</td>
</tr><tr>
  <td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
<b><font size="1">
  PACKAGE
</font></b></font></div>
</td>
  <TD align="left">
    <%if not l_action then %>
      <% strSQL = "select PACKAGE_NAME from PROD_PANEL_PACKAGE WHERE Product_id =" &
Session("PRODUCT_ID")
      objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
      showQueryListBox objRST, "P_PANEL_PKG" & i, 6, "SELECT PACKAGE",
l_panel_pkg
      objRST.close
    %>
    <% else
      Response.Write l_panel_pkg
    %>
  </td>
</tr>
```

```

end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
EXPENSE SET
</font></b></font></div>
</td>
<TD align="left">
<%if not l_action then %>
<% strSQL = "select e_set_name from EXPENSE_SETS where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID")
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "p_e_SET_NAME" & i,6, "SELECT EXPENSE SET", l_e_SET_NAME
objRST.close %>
<% else
Response.Write l_e_set_name
end if %>
</td>
</tr><tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
REVENUE SET
</font></b></font></div>
</td>
<TD align="left">
<%if not l_action then
strSQL = "select r_set_name from REVENUE_SETS where " _
& " PRODUCT_ID=" & Session("PRODUCT_ID")
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
showQueryListBox objRST, "p_r_SET_NAME",6, "SELECT REVENUE SET",
l_R_SET_NAME
objRST.close
else
Response.Write l_R_SET_NAME
end if %>
</td>
</tr>
<tr>
<td bgcolor="003366" height="20" width="140" align="center">
<div align="center">
<font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
<b><font size="1">
NEWSSTAND SALE
</font></b></font></div>
</td>
<TD align="left">
<%if not l_action then
dim vals
vals = "N,Y"
showLOVListBox vals, "P_NEWSTAND_SALE_MARKER", 2, "----",l_newstand_sale_marker
%>
<% else
Response.Write l_panel_date_start
end if %>
</td>
</tr>

```





editpanelpackage.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%
    l_ColumnCount = 2
    l_rec_id = Request.QueryString("Z")
    If Request.Form("UPDATE.x") <> "" then
        l_action = "Update"

    elseif Request.form("DELETE.x") <> "" then
        l_action = "Delete"

    end if
    l_sp_edit = "sp_Edit_Prod_Panel_Package"
    dim l_col_names(2)
    dim l_col_values(2)
    l_Title = " Panel Package "
    l_setupTabLink = "setupPanelPackage.asp"
    l_showPrevSelection= false
    l_file = "editPanelPackage.asp"

    l_col_names(0) = "PANEL PACKAGE"
    l_col_names(1) = "DESCRIPTION"
    l_query = "select Package_NAME, PACKAGE_DESCRIPTION " _
        & " FROM prod_panel_package " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then
    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query
        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if
end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id
'assign values to variables for setupTab's editMain
' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>

```

editpanelsubtype.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%
    l_ColumnCount = 2
    l_rec_id = Request.QueryString("Z")
    If Request.Form("UPDATE.x") <> "" then
        l_action = "Update"

    elseif Request.Form("DELETE.x") <> "" then
        l_action = "Delete"

    end if
    l_sp_edit = "sp_Edit_Corp_Panel_Sub_Type"
    dim l_col_names(2)
    dim l_col_values(2)
    l_Title = "Panel Sub Type"
    l_setupTabLink = "setupPanelSubType.asp"
    l_showPrevSelection = false
    l_file = "editPanelSubType.asp"

    l_col_names(0) = "PANEL SUB TYPE"
    l_col_names(1) = "DESCRIPTION"

    l_query = "select PANEL_SUB_TYPE_NAME, PANEL_SUB_TYPE_DESCR " _
        & " FROM corp_panel_sub_type " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric(l_rec_id) ) then

    if l_action = "Update" or l_action = "Delete" then

        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit

    else

        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if
end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>

```

editpaneltesttype.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%
    l_ColumnCount = 2
    l_rec_id = Request.QueryString("Z")
    If Request.Form("UPDATE.x") <> "" then
        l_action = "Update"

    elseif Request.Form("DELETE.x") <> "" then
        l_action = "Delete"

    end if
    l_sp_edit = "sp_Edit_Corp_Panel_Test_Type"
    dim l_col_names(2)
    dim l_col_values(2)
    l_Title = "Panel Test Type"
    l_setupTabLink = "setupPanelTestType.asp"
    l_showPrevSelection= false
    l_file = "editPanelTestType.asp"

    l_col_names(0) = "PANEL TEST TYPE"
    l_col_names(1) = "DESCRIPTION"

    l_query = "select PANEL_TEST_TYPE_NAME, PANEL_TEST_TYPE_DESCR " _
        & " FROM corp_panel_test_type " _
        & " WHERE rec_id = " & cint(l_rec_id)

    ' Response.Write l_rec_id
    If (IsNumeric(l_rec_id) ) then

        if l_action = "Update" or l_action = "Delete" then

            editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit

        else

            strSQL = l_query

            ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
            If (ObjRST.EOF = False) then
                l_col_values(0) = ObjRST.Fields(0)
                l_col_values(1) = ObjRST.Fields(1)
            end if
            objRST.close
        end if

        ' call setupTab's editmain to paint screen for user input
        editHeading l_file, l_file, l_rec_id

        'assign values to variables for setupTab's editMain

        ' call setupTab's editmain to paint screen for user input
        editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
        ' call setupTab's editBottom to close body, html, form
        editBottom %>
```



editpaneltype.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%
    l_ColumnCount = 2
    l_rec_id = Request.QueryString("Z")
    If Request.Form("UPDATE.x") <> "" then
        l_action = "Update"

    elseif Request.form("DELETE.x") <> "" then
        l_action = "Delete"

    end if
    l_sp_edit = "sp_Edit_Corp_Panel_Type"
    dim l_col_names(2)
    dim l_col_values(2)
    l_Title = "Panel Type"
    l_setupTabLink = "setupPanelType.asp"
    l_showPrevSelection= false
    l_file = "editPanelType.asp"

    l_col_names(0) = "PANEL TYPE"
    l_col_names(1) = "DESCRIPTION"

    l_query = "select PANEL_TYPE_NAME, PANEL_TYPE_DESCR " _
        & " FROM corp_panel_type " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then

    if l_action = "Update" or l_action = "Delete" then

        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit

    else

        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if

end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>
```

editpriorsrcdesc.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<%
dim l_col_names(8)
dim l_col_values(8)

l_ColumnCount = 8
l_rec_id = Request.QueryString("Z")
If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"

elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"

end if
l_rec_id = Request.QueryString("Z")
l_sp_edit = "sp_edit_prior_Source_desc"
l_file = "editPriorSrcDesc.asp"
l_Title = "Prior Source Description"
l_setupTabLink = "setupPriorSrcDesc.asp"
l_showPrevSelection= false
l_query = "select    prior_source_identifier, prior_source_desc, billing_cur_src_desc, model_source_cat,
model_source,agent_src_desc , valid_from_date , " _
    & " valid_to_date " _
    & " FROM Prior_source_desc " _
    & " WHERE rec_id = " & cint(l_rec_id)

if (l_action = "Update" or l_action = "Delete") and isnumeric(l_rec_id) then
    'editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    prior_source_identifier = request.form("P_0")
    prior_source_desc = request.form("P_1")
    billing_cur_src_desc = request.form("P_2")
    model_source_cat = request.form("P_3")
    model_source = request.form("P_4")
    agent_src_desc = request.form("P_5")
    valid_from_date = request.form("P_6")
    valid_to_date = request.form("P_7")

    'response.write(prior_source_identifier & "<br>")
    'response.write(prior_source_desc & "<br>")
    'response.write(billing_cur_src_desc & "<br>")
    'response.write(model_source_cat & "<br>")
    'response.write(model_source & "<br>")
    'response.write(agent_src_desc & "<br>")
    'response.write(valid_from_date & "<br>")
    'response.write(valid_to_date & "<br>")
    'response.write(l_action)
    'response.write(l_rec_id)

    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "sp_edit_prior_source_desc"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.Refresh
    objCMD.Parameters(1).Value = prior_source_identifier
    objCMD.Parameters(2).Value = prior_source_desc
    objCMD.Parameters(3).Value = billing_cur_src_desc
    objCMD.Parameters(4).Value = model_source_cat
    objCMD.Parameters(5).Value = model_source
    objCMD.Parameters(6).Value = agent_src_desc
    if valid_from_date > " " then objCMD.Parameters(7).Value = valid_from_date else objCMD.Parameters(7).value = null
    if valid_to_date > " " Then objCMD.Parameters(8).Value = valid_to_date else objcmd.Parameters(8).Value = null
    objCMD.Parameters(9).Value = l_rec_id

```

```

if l_action="Update" then objCMD.Parameters(10).Value = 0
if l_action="Delete" then objCMD.Parameters(10).Value = 1
Set TempNonSessionVar = objCMD.Execute()
Response.Redirect l_setuptablink
else
    strSQL = l_query
    ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    If (ObjRST.EOF = False) and isnumeric(l_rec_id) then
        prior_source_identfier = objRST("prior_source_identfier")
        prior_source_desc = objRST("prior_source_desc")
        billing_cur_src_desc = objRST("billing_cur_src_desc")
        model_source_cat = objRST("model_source_cat")
        model_source = objRST("model_source")
        agent_src_desc = objRST("agent_src_desc")
        valid_from_date = objRST("valid_from_date")
        valid_to_date = objRST("valid_to_date")
    end if
    objRST.close
end if

%>
<html>
<head>
<title>You are on editPriorSrcDesc.asp</title></head>
<body bgcolor="#99CCFF" vlink="000000" link="003366">
<form method="post" action="editPriorSrcDesc.asp?Z=<%=l_rec_id%>">

<table BORDER="0">
    <tr valign="top">
        <td align="left" width="640" colspan="3">
            
        </td>
    </tr>

    <tr valign="top">
        <td align="left" width="640" colspan="3">
            <table width="500" border="0" cellspacing="3" cellpadding="2">
                <tr valign="top">
                    <td height="25" align="left">
                        <div align="left"></div>
                    </td>
                    <td height="25" align="left">
                        <input type="IMAGE" Value="UPDATE" Name="UPDATE"
src="./graphics/submit1222.gif" width="54" height="16" border="0">
                        
                        <input type="IMAGE" Value="DELETE" Name="DELETE"
src="./graphics/delete1222.gif" width="53" height="16" border="0">
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <br>
                        <font face="Verdana, Arial, Helvetica, sans-serif" size="4">
                        <a href="setupPriorSrcDesc.asp">Return to Prior Source
                    </td>
                </tr>
            </table>
        </td>
    </tr>

    <tr>
        <td colspan="3">
            <div align="center">
                <font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">
                IDENTIFIER</font></b></font></div>
            </td>
        <td align="LEFT">
            <font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">

```

value="<%=prior_source_idenfier %>">	<input type="TEXT" NAME="P_0" id="P_0"
	</font>
	</td>
	</tr>
	<tr>
	<td bgcolor="003366" height="20" width="140" align="center">
	<div align="center">
	<font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">	
DESC</font></b></font></div>	<b><font size="1">RENEWAL PRIOR SOURCE
	</td>
	<td align="LEFT">
	<font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">	
value="<%=prior_source_desc%>">	<input type="TEXT" NAME="P_1" id="P_1"
	</font>
	</td>
	</tr>
	<tr>
	<td bgcolor="003366" height="20" width="140" align="center">
	<div align="center">
	<font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">	
DESC</font></b></font></div>	<b><font size="1">BILLINGS CURRENT SOURCE
	</td>
	<td align="LEFT">
	<font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">	
value="<%=billing_cur_src_desc%>">	<input type="TEXT" NAME="P_2" id="P_2"
	</font>
	</td>
	</tr>
	<tr>
	<td bgcolor="003366" height="20" width="140" align="center">
	<div align="center">
	<font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">	
CATEGORY</font></b></font></div>	<b><font size="1">MODEL SOURCE
	</td>
	<td align="LEFT">
	<font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">	
	<SELECT name="P_3" id="P_3">
	<% strSQL = "select src_cat_name from src_category where
product_id = " & Session("PRODUCT_ID") & " order by src_cat_name"	ObjRST.open strSQL, strSQLOLEDB, adOpenStatic,
adLockOptimistic	
	while not objRST.eof
	response.write("<OPTION ")
selected ")	if objRST(0) = model_source_cat then response.write ("
objRST(0) & "<OPTION>")	response.write (" value=" & objRST(0) & ">" &
	objRST.movenext
	wend
	objRST.close
	%>
value="<%=model source cat%>">	<%><input type="TEXT" NAME="P_3"



		</font>
	</tr>	
	<tr>	
	<td bgcolor="003366" height="20" width="140" align="center"> <div align="center"> <font face="Verdana, Arial, Helvetica, sans-serif"	
color="#FFFFFF">		
DESCRIPTION</font></b></font></div>		<b><font size="1">MODEL SOURCE
	</td>	
color="#FFFFFF">	<td align="LEFT">	<font face="Verdana, Arial, Helvetica, sans-serif"
value="<%=model_source%>">		<input type="TEXT" NAME="P_4" id="P_4"
		</font>
	</td>	
	</tr>	
	<tr>	
	<td bgcolor="003366" height="20" width="140" align="center"> <div align="center"> <font face="Verdana, Arial, Helvetica, sans-serif"	
color="#FFFFFF">		
DESCRIPTION</font></b></font></div>		<b><font size="1">AGENT SOURCE
	</td>	
color="#FFFFFF">	<td align="LEFT">	<font face="Verdana, Arial, Helvetica, sans-serif"
value="<%=agent_src_desc%>">		<input type="TEXT" NAME="P_5"
		</font>
	</td>	
	</tr>	
	<tr>	
	<td bgcolor="003366" height="20" width="140" align="center"> <div align="center"> <font face="Verdana, Arial, Helvetica, sans-serif"	
color="#FFFFFF">		
DATE</font></b></font></div>		<b><font size="1">VALID FROM
	</td>	
color="#FFFFFF">	<td align="LEFT">	<font face="Verdana, Arial, Helvetica, sans-serif"
value="<%=valid_from_date%>">		<input type="TEXT" NAME="P_6"
		</font>
	</td>	
	</tr>	
	<tr>	
	<td bgcolor="003366" height="20" width="140" align="center"> <div align="center"> <font face="Verdana, Arial, Helvetica, sans-serif"	
color="#FFFFFF">		
		<b><font size="1">VALID TO DATE</font></b></font></div>
	</td>	
color="#FFFFFF">	<td align="LEFT">	<font face="Verdana, Arial, Helvetica, sans-serif"
value="<%=valid_to_date%>">		<input type="TEXT" NAME="P_7"
		</font>



```

                                </td>
                                </tr>
                                </table>
                                </TD>
                                </TR>
                                </TABLE>
                                </form>
                                </body>
                                </html>

```

editrset.asp

```

<%@ Language=VBScript %>

<!-- #include file="..support/global_constants.asp" -->
<!-- #include file="..support/dbConnection.asp" -->
<!-- #include file="..support/adoUtils.asp" -->
<!-- #include file="..support/srcTab.asp" -->
<!-- #include file="..support/pickListObj.asp" -->
<!-- #include file="..support/JavaScripts.asp" -->

<%

dim DEBUG
DEBUG = false

dim l_label(10)
dim l_values(33)
'dim l_basis_opt(10)(7)
    l_label(0) = "Sub Revenue"
    l_label(1) = "Ad Revenue"
    l_label(2) = "List Rental Revenue"
    l_label(3) = "Other Revenue"

l_rec_ID = Request.QueryString("Z")

If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"
elseif Request.Form("DELETE.x") <> "" then
    l_action = "Delete"
else
    l_action = "x"
end if

'Response.Write "Hate " & l_action
If l_action = "Update" or l_action = "Delete" then
    ' Response.Write "Save or Delete Invoked"
    'if Request.Form("P_REC_ID") > "" then

        l_rec_id = Request("P_REC_ID")
        ' Response.Write "<BR>Rec id=" & l_rec_id & "=" & Request("P_REC_ID") & "<BR>"
        for i = 0 to 7
            l_values(i) = Request.Form("P_" & i)
        next
        l_r_set_name = Request.Form("P_R_SET_NAME")
        'onse.Write "<BR>Rec id " & Request.Form("P_R_SET_NAME") & "<BR>"
        if l_r_set_name > "" then
            objCMD.ActiveConnection = strSQLOLEDB
            objCMD.CommandText = "sp_edit_prod_r_set"
            objCMD.CommandType = adCmdStoredProc
            objCMD.Parameters.Refresh
            '            USER SUPPLIED
            objCMD.Parameters(1).Value = Session("PRODUCT_ID")
            objCMD.Parameters(2).Value = l_r_set_name
            for k = 0 to 7
                IF DEBUG = TRUE THEN
                    ' Response.Write (k + 3) & " " & k & " " & l_values(k) & "<br>"
                END IF
            ' Response.Write "<BR>l_value " & l_values(k) & "<BR>"
            if k mod 2 = 0 then
                objCMD.Parameters(k + 3).Value = validateNumber(l_values(k))
            else
                objCMD.Parameters(k + 3).Value = validatePickList(l_values(k))
            end if
        end for
    end if
end if

```

```

        end if
        next
        Response.Write k
        Response.Write "<BR>Rec id " & l_rec_id & "<BR>"
        objCMD.Parameters(11).Value = l_rec_id

        if l_action = "Delete" then
            objCMD.Parameters(12).Value = 1
        else
            objCMD.Parameters(12).Value = 0
        end if

        Set TempNonSessionVar = objCMD.Execute()

    end if
end if

Response.Redirect "setupRset.asp"
else
    l_action = "Edit"
    Response.Write "Edit Invoked"
    strSQL = "select " & "
        & " product_id r0, " & "
        & " r_set_name r1, " & "
        & " subs_revenue r2, " & "
        & " subs_cpm_cpu r3, " & "
        & " ad_revenue r4, " & "
        & " ad_cpm_cpu r5, " & "
        & " listrental_revenue r6, " & "
        & " listrental_cpm_cpu r7, " & "
        & " other_revenue r8, " & "
        & " other_cpm_cpu r9 " & "
        & " from revenue_sets " & "
        & " where rec_id = " & l_rec_id

    Response.Write strSQL
    ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
    l_product_id = ObjRST.fields(0)
    l_r_set_name = ObjRST.fields(1)
    for i = 0 to 7
        l_values(i) = ObjRST.fields(i + 2)
        Response.Write i + 1 & " "
    next

    ObjRST.close

end if
Response.Write "Hate " & l_action
%>
<HTML>
<HEAD>
<title>editRset.asp
</title>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<script language="JavaScript">
<!--
function MM_swapImgRestore() { //v2.0
    if (document.MM_swapImgData != null)
        for (var i=0; i<(document.MM_swapImgData.length-1); i+=2)
            document.MM_swapImgData[i].src = document.MM_swapImgData[i+1];
}

function MM_preloadImages() { //v2.0
    if (document.images) {
        var imgFiles = MM_preloadImages.arguments;
        if (document.preloadArray==null) document.preloadArray = new Array();
        var i = document.preloadArray.length;
        with (document) for (var j=0; j<imgFiles.length; j++) if (imgFiles[j].charAt(0)!="#"){

```



```
preloadArray[i] = new Image;
preloadArray[i++].src = imgFiles[j];
} }
}

function MM_swapImage() { //v2.0
var i,j=0,objStr,obj.swapArray=new Array,oldArray=document.MM_swapImgData;
for (i=0; i < (MM_swapImage.arguments.length-2); i+=3) {
objStr = MM_swapImage.arguments[(navigator.appName == 'Netscape')?i:i+1];
if ((objStr.indexOf('document.layers[')===0 && document.layers===null) ||
(objStr.indexOf('document.all[') ===0 && document.all ===null))
objStr = 'document'+objStr.substring(objStr.lastIndexOf('.'),objStr.length);
obj = eval(objStr);
if (obj != null) {
swapArray[j++] = obj;
swapArray[j++] = (oldArray===null || oldArray[j-1]!==obj)?obj.src:oldArray[j];
obj.src = MM_swapImage.arguments[i+2];
} }
document.MM_swapImgData = swapArray; //used for restore
}
//-->
</script>
<body bgcolor="#99CCFF" vlink="#000000" link="#003366">

<form method="post" action="editRSet.asp" id=form1 name=form1>
<% If l_rec_ID > " " then %>
<input type="HIDDEN" name="P_REC_ID" value="<% Response.Write l_rec_id %>">
<% end if %>
<TABLE BORDER=0 width="311" >

<tr valign="top">
<td align="left" colspan="3" height="20">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="25%"></td>
<td width="35%">&nbsp;</td>
<td width="70" align="right">
<div align="right"> </div>
</td>
<td width="70" align="right">
<div align="right"> </div>
</td>
</tr>
<tr>
<td width="25%">&nbsp;</td>
<td width="35%">&nbsp;</td>
<td width="70" align="right">
<input type="IMAGE" value="UPDATE" name="UPDATE"

src="./graphics/submit1222.gif" width="54" height="16" border="0">

</td>
<td width="70" align="right">
<input type="IMAGE" value="DELETE" name="DELETE"

src="./graphics/delete1222.gif" width="53" height="16" border="0">

</td>
</tr>
<tr>
<td width="25%">&nbsp;</td>
<td width="35%" colspan="3" align="right" height="24" valign="bottom">
<div align="right"><font face="Verdana, Arial, Helvetica, sans-serif" size="4"><a href="setupRset.asp">Return
to Revenue Set Setup Table</a></font></div>
</td>
</tr>
</table>
</td>
</tr>
</table>
```

```

<tr>
  <td width="150" align="center" height="10">&nbsp;</td>
  <td align="LEFT" height="10">&nbsp;</td>
</tr>
<tr>
  <td bgcolor="003366" height="20" width="109" align="center">
    <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
      <b><font size="1">
        Revenue Set
      </font></b></font></td>
  <td colspan="2">
    <input type="TEXT" name='p_r_set_name'

      id=0 value='<%Response.Write l_r_set_name %>' size="27">
    </td>
  </tr>
  <%
    for i=0 to 3
  %>
  <tr>
    <td bgcolor="003366" height="20" width="109" align="center">
      <font face="Verdana, Arial, Helvetica, sans-serif" color="FFFFFF">
        <b><font size="1">
          <%
            Response.Write l_label(i)
          %>
        </font></b></font></td>
      </td>
      <%
        for j = 0 to 1
          in_cnt = i * 2 + j
        %>

        <td>
          <% if j = 1 then %>
            <SELECT name='P_<%Response.write in_cnt %>'>
              <% if l_values(in_cnt) = "CPM" then %>
                <OPTION value=0>CPM or CPU
                <OPTION value='CPM' selected>CPM</option>
                <OPTION value='CPU'>CPU</option>
              <% elseif l_values(in_cnt) = "CPU" then %>
                <OPTION value=0>CPM or CPU
                <OPTION value='CPM'>CPM</option>
                <OPTION value='CPU' selected>CPU</option>
              <% else %>
                <OPTION value=0>CPM or CPU
                <OPTION value='CPM'>CPM</option>
                <OPTION value='CPU'>CPU</option>
              <% END IF %>
            </Select>
            <% elseif j = 2 then %>

            <% else %>
              <input type="TEXT" NAME='P_<%Response.write in_cnt %>'
                <%if l_action="Edit" then %>
                  value='<%Response.Write l_values(in_cnt) %>'
                <%end if %>
                size =12>
              <% end if %>
            </td>
            <% next %>
          </td>
        <% next %>
      </tr>
    <% next %>
  </table>
  <p>&nbsp;</p>
</form>
</BODY>

```



</HTML>

editsource.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->

<%

If Request.Form("UPDATE.x") <> "" then
    l_button = "Update"
elseif Request.Form("DELETE.x") <> "" then
    l_button = "Delete"
end if

l_rec_id = Request.QueryString("Z")
Response.Write l_rec_id
If (IsNumeric( l_rec_id ) ) then
    if l_button = "Update" or l_button = "Delete" then
        l_action = true
        l_source = Request.Form("P_SOURCE")
        l_src_ind = Request.Form("P_SRC_IND")

        objCMD.ActiveConnection = strSQLOLEDB
        objCMD.CommandText = "sp_edit_source"
        objCMD.CommandType = adCmdStoredProc
        objCMD.Parameters.Refresh
        objCMD.Parameters(1).Value = l_source
        objCMD.Parameters(2).Value = l_src_ind
        objCMD.Parameters(3).Value = Session("U")
        objCMD.Parameters(4).Value = cint(l_rec_id)
        if l_button = "Delete" then
            objCMD.Parameters(5).Value = 1
        else
            objCMD.Parameters(5).Value = 0
        end if
        Set TempNonSessionVar = objCMD.Execute()
        Response.Redirect "admin4a.asp?display=1"

        l_action = true
        l_source = Request.Form("P_SOURCE")
        l_src_ind = Request.Form("P_SRC_IND")

    else
        l_action = false

        strSQL = "select SOURCE_NAME, SRC_INDICATOR from SOURCE where " _
            & " rec_id = " & l_rec_id
        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_source = ObjRST.Fields("SOURCE_NAME")
            l_src_ind = ObjRST.Fields("SRC_INDICATOR")
        end if
        ObjRST.close
    end if
end if

%>
<html>
<head>
<title>EditSource.asp</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366">
```



```

<form method="post" action="editSource.asp?Z=<%Response.Write l_rec_id %>" id=form1 name=form1>
<table width="640" border="0" cellpadding="2">
  <% prevSelections(3) %>

  <tr valign="top">
    <td align="left" width="640" colspan="3"></td>
  </tr>
  <tr valign="top">
    <td align="left" width="640" colspan="3">
      <table width="500" border="0" cellspacing="3" cellpadding="2">
        <tr valign="top">
          <td height="25" align="left">
            <div align="left"></div>
          </td>
          <td height="25" align="left">
            <input type='IMAGE' Value="UPDATE" Name="UPDATE"
              src="..\graphics/submit1222.gif" width="54" height="16" border="0">
            
            <%> <input type='IMAGE' Value="DELETE" Name="DELETE"
              'src="..\graphics/delete1222.gif" width="53" height="16" border="0"> %>
            <font face="Verdana, Arial, Helvetica, sans-serif" size="4">
              <a href="admin4a.asp">Return
                to Source</a></font></td>
            </tr>
            <tr>
              <td bgcolor="#003366" height="20" width="140" align="center">
                <div align="center">
                  <font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">
                    <b><font size="1">
                      SOURCE NAME
                    </font></b></font></div>
                </td>
                <td bgcolor="#99CCFF" height="20" width="360" align="left">
                  <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">
                    <%if not l_action then
                      strSQL = "select U_SOURCE_NAME FROM Source_descriptions "
                      objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
                      showQueryListBox objRST, "P_SOURCE", 6, "SELECT SOURCE", l_source
                      objRST.close
                    else
                      Response.Write l_source
                    end if %>
                  </font></b></font></div>
                </td>
            </tr>
            <tr>
              <td width="140" bgcolor="#003366" height="20" align="center">
                <div align="center"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1">SOURCE
                  INDICATOR </font></b></font></div>
              </td>
              <td width="360" bgcolor="#99CCFF" height="20" align="left">
                <div align="left"><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF"><b><font size="1"
color="#000000">

                  <%if not l_action then %>
                  <input type="TEXT" NAME="P_SRC_IND"
                    id=0 value='<%Response.Write l_src_ind %>' maxlength="30" size="6">
                  <% else
                    Response.Write l_src_ind
                  end if %>
                  </TD>

                </font></b></font></div>
              </td>
            </tr>

```



```
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

editsourceName.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<%
dim l_col_names(2)
dim l_col_values(2)

l_ColumnCount = 2
l_rec_id = Request.QueryString("Z")
If Request.Form("UPDATE.x") <> "" then
    l_action = "Update"

elseif Request.form("DELETE.x") <> "" then
    l_action = "Delete"

end if
l_sp_edit = "sp_Edit_Source_Description"
l_file = "editSourceName.asp"
l_Title = "Source Description"
l_setupTabLink = "setupSourceName.asp"
l_showPrevSelection= false

l_col_names(0) = "SOURCE "
l_col_names(1) = "DESCRIPTION"

l_query = "select U_SOURCE_NAME,    U_source_description " _
        & " FROM Source_descriptions " _
        & " WHERE rec_id = " & cint(l_rec_id)

' Response.Write l_rec_id
If (IsNumeric( l_rec_id) ) then

    if l_action = "Update" or l_action = "Delete" then
        editAccept l_rec_id, l_action, l_ColumnCount, l_setupTabLink, l_sp_edit
    else
        strSQL = l_query

        ObjRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
        If (ObjRST.EOF = False) then
            l_col_values(0) = ObjRST.Fields(0)
            l_col_values(1) = ObjRST.Fields(1)
        end if
        objRST.close
    end if

end if

' call setupTab's editmain to paint screen for user input
editHeading l_file, l_file, l_rec_id

'assign values to variables for setupTab's editMain

' call setupTab's editmain to paint screen for user input
editMain l_Title, l_setupTabLink, l_showPrevSelection, l_col_names, l_col_values, l_ColumnCount
' call setupTab's editBottom to close body, html, form
editBottom %>

```

ms\_editcampaign.asp

```
<%@ Language="VBScript" %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/navigation.asp" -->
<!-- #include file="../support/objedit.asp" -->
<%
l_this_file = "ms_editcampaign.asp"
Session("refer_page") = "admin5a.asp"
if (Request.Form("setupTab") <> "0" and Request.Form("setupTab") > " ") then Response.Redirect(Request.Form("setupTab"))

savechanges = False
page = request.querystring("pg")
if not (len(page) > 0) then
    page = request.form("pg")
    if request.form("groupedit") > " " then savechanges = True
end if
toomanyrecords = request.querystring("ro")
'Update
if savechanges = True then
    'response.write ("Entering ...")
    'Get records
    recordcount = Request.form("p_RecCount")
    recordid = ""
    recordidlist = ""
    'response.write (" RecordCount -> " & cstr(recordcount) & "<br>")
    'response.write("There are " & cstr(recordcount) & " records <br>")

    for i = 1 to recordcount
        recordid = Request.form("rec_id" & cstr(i))
        'response.write ("<br>Record id: " & recordid)
        if recordid > " " then

            'init obj
            set PGE = new objPostGroupEdit
            PGE.init
            PGE.MSE_Row = i
            PGE.AddRecord recordid
            'response.write(cstr(recordid) + "<br>")

            'Set Stored Procedure info
            PGE.ClassType = "Multi Edit"
            PGE.SP = "sp_edit_campaign"
            PGE.TableName = "campaigns"
            if Request.form("GroupEdit") > " " then PGE.IsDelete = 0
            if request.form("deletebutton") > " " then PGE.IsDelete = 1
            PGE.AddParam "camp_name"
            PGE.AddParam "camp_date_fr"
            PGE.AddParam "camp_type"
            PGE.AddParam "camp_status_name"
            PGE.AddParam "e_set_name"
            PGE.AddParam "r_set_name"
            'PGE.AddParam "newstand_sale"
            PGE.AddParam "budgeted_gross_pct"
            PGE.AddParam "budgeted_vol"
            PGE.AddParam "login"
            PGE.AddParam "rec_id"
            PGE.AddParam "IsDelete"
            objCMD.ActiveConnection = strSQLOLEDB
            objCMD.CommandText = "sp_edit_campaign"
            objCMD.CommandType = adCmdStoredProc
            objCMD.parameters.refresh
            PGE.PostGroupEditsBySP
            'response.write ("Changes Saved")
```



```
        end if
    next i
end

if
    'Multi Select
    dim MSE                                ' Multi Select object
    dim strSQL                            ' SQL Query text
    dim NumOfRecords ' Total records matching whereclause - used for Prev and Next buttons
    dim RecsPerPage    ' max number of records to display on each page

    'init obj
    set MSE = new objGroupEdit
    MSE.init
    MSE.ClassType = "MultiSelect"
    MSE.action = l_this_file
    MSE.Return_Page = "admin5a.asp"
    MSE.TableHeader = "Multi Edit"
    MSE.show = true
    recordidlist = Session("RECORDIDLIST")
    MSE.MS_Page = page
    RecsPerPage = 5
    MSE.MS_RecsPerPage = RecsPerPage
    MSE.MS_Table = "Campaigns"
    MS_WhereClause = " where grp_name = " & Session("GRP_NAME") & " and PRODUCT_ID=" & Session("PRODUCT_ID") & "
    and src_cat_name = " & Session("SRC_CAT_NAME") & " and source_name = " & Session("SOURCE_NAME") & " and
    rec_id in (" & Session("RECORDIDLIST") & ")"
    MS_OrderBy = " Order by camp_name desc "
    'response.write MS_WhereClause

    'open recordset for current page
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "PagedQuery"
    objCMD.CommandType = adCmdStoredProc
    objCMD.parameters.refresh
    objCMD.Parameters("@Page").value = page
    objCMD.Parameters("@RecsPerPage") = RecsPerPage
    objCMD.Parameters("@QueryTable") = "Campaigns"
    objCMD.Parameters("@WhereClause") = MS_WhereClause
    objCMD.Parameters("@OrderBy") = MS_OrderBy
    set objRST = objCMD.execute()
    'response.write("Record ID list > " & Session("RECORDIDLIST"))
    'response.write ("Page ->" + cstr(Request.querystring("pg")))
    'strSQL = "pagedquery " + cstr(page) + ",5,campaigns," + chr(34) + MS_WhereClause + Chr(34) + "," + chr(34) + MS_OrderBy +
    chr(34)
    'response.write (strSQL)
    'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

    NumOfRecords = 0
    if not objRST.eof then
        NumOfRecords = objRST("MoreRecords")
        MSE.MS_NumOfRecords = NumOfRecords
    else
        'There are no more records
        'response.write "There are no more records to edit"
        response.redirect "admin5a.asp"
    end if
    while not objRST.eof
        MSE.NextRow

        'Campaign Name
        index = MSE.AddItem
        MSE.fieldType index, "TEXT"
        MSE.ColumnHeader index, "CAMPAIGN NAME"
        MSE.default index, mycstr(objRST("camp_name"))

        'Campaign Start Date
        index = MSE.AddItem
        MSE.fieldType index, "TEXT"
```



```
MSE.ColumnHeader index, "CAMPAIGN START DATE"
MSE.default index, mycstr(objRST("camp_date_fr"))

'Campaign Type
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "CAMPAIGN TYPE"
MSE strSQL index, "select CAMP_TYPE_NAME from PROD_CAMP_TYPE where PRODUCT_ID=" &
Session("PRODUCT_ID") & " order by camp_type_name "
MSE.default index, mycstr(objRST("camp_type"))

'Campaign Status
index = MSE.AddItem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "CAMPAIGN STATUS"
MSE strSQL index, "select CAMP_STATUS_NAME from PROD_CAMP_STATUS where PRODUCT_ID=" &
Session("PRODUCT_ID") & " order by camp_status_name "
MSE.default index, mycstr(objRST("camp_status_name"))

'Expense Set
index = MSE.AddItem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "EXPENSE SET"
MSE strSQL index, "select e_set_name from EXPENSE_SETS where PRODUCT_ID=" & Session("PRODUCT_ID") & "
order by e_set_name "
MSE.default index, mycstr(objRST("e_set_name"))

'Revenue Set
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "REVENUE SET"
MSE strSQL index, "SELECT r_set_name from revenue_sets where product_id = " & Session("PRODUCT_ID") & "
order by r_set_name"
MSE.default index, mycstr(objRST("r_set_name"))

'Newstand Sale
index = MSE.AddItem
MSE.fieldType index, "TEXT"
MSE.ColumnHeader index, "NEWSTAND SALE"
MSE.default index, mycstr(objRST("newstand_sale"))

'Budgeted Gross %
index = MSE.AddItem
MSE.fieldtype index, "TEXT"
MSE.ColumnHeader index, "BUDGETED GROSS %"
MSE.default index, mycstr(objRST("budgeted_gross_pct"))

'Budgeted Vol
index = MSE.AddItem
MSE.fieldtype index, "TEXT"
MSE.ColumnHeader index, "BUDGETED VOL"
MSE.default index, mycstr(objRST("budgeted_vol"))

'Record ID
index = MSE.AddItem
MSE.fieldType index, "RECORDID"
MSE.ColumnHeader index, ""
MSE strSQL index, "N,Y"
MSE.default index, mycstr(objRST("rec_id"))
objRST.movenext
wend ' objRST
objRST.close

'Table Headers
%>      <form method="POST" action="<%=l_this_file%>" name="Group_Edit_Form" cellpadding="3" cellspacing="3">
          <table>

<%
```



```
'response.write (Session("RECORDIDLIST") & " Hello World")
```

```
'Set record count
```

```
%> <input type="HIDDEN" value="<%=RecsPerPage%>" name="p_RecCount"> <%=
```

```
'Display Table
```

```
MSE.DisplayEditTable
```

```
%>
```

ms\_editkey.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/navigation.asp" -->
<!-- #include file="..\support/objedit.asp" -->
<%
l_this_file="ms_editkey.asp"
Session("refer_page")="ms_editkey.asp"
if (Request.Form("setupTab")<>"0" and Request.Form("setupTab")>" ") then Response.Redirect(Request.Form("setupTab"))

savechanges=False
page=request.querystring("pg")
if not len(page)>0 then
    page=request.form("pg")
    savechanges=True
end if

'Update
if savechanges=True then

    'Get records
    recordcount=Request.form("p_RecCount")
    'response.write (" RecordCount -> " & cstr(recordcount) & "<br>")
    recordid=""
    recordidlist=""

    'response.write("There are " & cstr(recordcount) & " records <br>")
    for i=1 to recordcount
        recordid=Request.form("rec_id" & cstr(i))
        if recordid>" " then

            'init obj
            set PGE=new objPostGroupEdit
            PGE.init
            PGE.MSE_Row=i
            PGE.AddRecord recordid
            'response.write(cstr(recordid) + "<br>")

            'Set Stored Procedure info
            PGE.ClassType="Multi Edit"
            PGE.SP="sp_edit_key"
            PGE.TableName="keys"
            PGE.IsDelete=0
            PGE.AddParameter "p_key_descr"
            PGE.AddParameter "mail_qty_override"
            PGE.AddParameter "newstand_sale"
            PGE.AddParameter "newstand_sale_marker"
            PGE.AddParameter "list_cat_name"
            PGE.AddParameter "list_name"
            PGE.AddParameter "list_segment_name"
            PGE.AddParameter "list_cost_actual_cpm"
            PGE.AddParameter "list_cost_basis"
            PGE.AddParameter "list_cost_factor"
            PGE.AddParameter "merge_purge_qty"
            PGE.AddParameter "e_set_name"
            PGE.AddParameter "r_set_name"
            PGE.AddParameter "panel_pkg"
            PGE.AddParameter "subtotal_flag"
            PGE.AddParameter "optional_effort_id"
            PGE.AddParameter "rec_id"
            PGE.AddParameter "IsDelete"
            objCMD.ActiveConnection=strSQLOLEDB
            objCMD.CommandText="sp_edit_key"
            objCMD.CommandType=adCmdStoredProc

```





```
objCMD.parameters.refresh
PGE.PostGroupEditsBySP
    end if
next 'i
end    if

'Multi Select
dim MSE                                ' Multi Select object
dim strSQL                            ' SQL Query text
dim NumOfRecords ' Total records matching whereclause - used for Prev and Next buttons
dim RecsPerPage    ' max number of records to display on each page

'init obj
set MSE = new objGroupEdit
MSE.init
MSE.action = l_this_file
MSE.ClassType = "MultiSelect"
MSE.TableHeader = "Multi Edit"
MSE.show = true
recordidlist = Session("RECORDIDLIST")
MSE.MS_Page = page
RecsPerPage = 5
MSE.MS_RecsPerPage = RecsPerPage
MSE.MS_Table = "keys"
MSE.Return_Page = "adminkeysright.asp"
'MS_WhereClause = " where grp_name = " & Session("GRP_NAME") & " and PRODUCT_ID=" & Session("PRODUCT_ID") & "
and src_cat_name = " & Session("SRC_CAT_NAME") & " and source_name = " & Session("SOURCE_NAME") & " and "
and Camp_name = " & Session("CAMP_NAME") & " and rec_id in (" & Session("RECORDIDLIST") & ")"
MS_WhereClause = " where rec_id in (" & Session("RECORDIDLIST") & ")"
MS_OrderBy = " Order by full_promo_key "
'response.write MS_WhereClause

'open recordset for current page
objCMD.ActiveConnection = strSQLOLEDB
objCMD.CommandText = "PagedQuery"
objCMD.CommandType = adCmdStoredProc
objCMD.parameters.refresh
objCMD.Parameters("@Page").value = page
objCMD.Parameters("@RecsPerPage") = RecsPerPage
objCMD.Parameters("@QueryTable") = "keys"
objCMD.Parameters("@WhereClause") = MS_WhereClause
objCMD.Parameters("@OrderBy") = MS_OrderBy
'response.write ("Page ->" + cstr(Request.querystring("pg")))
'strSQL = "pagedquery " + cstr(page) + ",5,'keys'," & chr(34) + MS_WhereClause & Chr(34) + "," & chr(34) + MS_OrderBy & chr(34)
'response.write (strSQL)
set objRST = objCMD.execute()
'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

NumOfRecords = 0
if not objRST.eof then
    NumOfRecords = objRST("MoreRecords")
    MSE.MS_NumOfRecords = NumOfRecords
'response.write (NumOfRecords)
else
    'There are no more records
'response.write "There are no more records to edit"
response.redirect "adminkeysright.asp"
end if

while not objRST.eof
    MSE.NextRow

    'Key
    index = MSE.AddItem
    MSE.fieldType index, "NOTAPARAMETER"
    MSE.ColumnHeader index, "KEY"
    MSE.default index, mycstr(objRST("full_promo_key"))
```

```
'Key Description
index = MSE.AddItem
MSE.fieldType index, "TEXT"
MSE.ColumnHeader index, "KEY DESCRIPTION"
MSE.default index, mycstr(objRST("p_key_descr"))
```

```
'Mail Qty
index = MSE.AddItem
MSE.fieldType index, "NOTAPARAMETER"
MSE.ColumnHeader index, "MAIL QTY"
MSE.default index, mycstr(objRST("key_mail_qty"))
```

```
'Mail Qty OverRide
index = MSE.AddItem
MSE.fieldtype index, "TEXT"
MSE.ColumnHeader index, "MAIL QTY OVERRIDE/DRAW"
MSE.default index, mycstr(objRST("mail_qty_override"))
```

```
'Newstand Sale
index = MSE.AddItem
MSE.fieldtype index, "TEXT"
MSE.ColumnHeader index, "NEWSTAND SALE"
MSE.default index, mycstr(objRST("Newstand_Sale"))
```

```
'Newstand Sale Marker
index = MSE.AddItem
MSE.fieldtype index, "SELECTLOV"
MSE.ColumnHeader index, "RESPONSE BASED ON SALE (Y), OR DRAW (N)"
MSE.strSQL index, "N,Y"
MSE.default index, mycstr(objRST("Newstand_Sale_Marker"))
```

```
'List Category
index = MSE.AddItem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "LIST CATEGORY"
MSE.strSQL index, "SELECT list_cat_name from list_cats where product_id = " & Session("PRODUCT_ID") & " order
by list_cat_name"
MSE.default index, mycstr(objRST("list_cat_name"))
```

```
'List Name
index = MSE.AddItem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "LIST NAME"
MSE.strSQL index, "SELECT list_name from list_names where product_id = " & Session("PRODUCT_ID") & " order by
list_name"
MSE.default index, mycstr(objRST("list_name"))
```

```
'List Segment Name
index = MSE.AddItem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "LIST SEGMENT NAME"
MSE.strSQL index, "SELECT list_segment_name from list_segments where product_id = " & Session("PRODUCT_ID")
& " order by list_segment_name"
MSE.default index, mycstr(objRST("list_segment_name"))
```

```
'List cost actual CPM
index = MSE.AddItem
MSE.fieldType index, "TEXT"
MSE.ColumnHeader index, "LIST COST ACTUAL CPM"
MSE.default index, mycstr(objRST("list_cost_actual_cpm"))
```

```
'List Cost Basis
index = MSE.AddItem
MSE.fieldType index, "SELECTLOV"
MSE.ColumnHeader index, "LIST COST BASIS"
MSE.strSQL index, "Mail QTY,Merge Purge QTY,Undefined"
MSE.default index, mycstr(objRST("list_cost_basis"))
```

```

'List Cost Factor
index = MSE.AddItem
MSE.FieldType index, "TEXT"
MSE.ColumnHeader index, "LIST COST FACTOR"
MSE.default index, mycstr(objRST("list_cost_factor"))

'Merge Purge QTY
index = MSE.AddItem
MSE.Fieldtype index, "TEXT"
MSE.ColumnHeader index, "MERGE PURGE ATY"
MSE.default index, mycstr(objRST("merge_purge_qty"))

'Expense Set
index = MSE.additem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "EXPENSE SET"
MSE strSQL index, "select e_set_name from expense_sets where product_id = " & Session("Product_id") & " order by
e_set_name"
MSE.default index, mycstr(objRST("e_set_name"))

'Revenue Set
index = MSE.additem
MSE.fieldtype index, "SELECT"
MSE.ColumnHeader index, "REVEUE SET"
MSE strSQL index, "select r_set_name from revenue_sets where product_id = " & Session("PRODUCT_ID") & " order by
r_set_name"
MSE.default index, mycstr(objRST("r_set_name"))

'Package
index = MSE.AddItem
MSE.fieldtype index, "SELECT"
MSE.Columnheader index, "PACKAGE"
MSE strSQL index, "select package_name from prod_panel_package where product_id = " & Session("PRODUCT_ID")
& " order by package_name"
MSE.default index, mycstr(objRST("panel_pkg"))

'Subtotal Flag
index = MSE.AddItem
MSE.fieldType index, "SELECTLOV"
MSE.ColumnHeader index, "SUBTOTAL FLAG"
MSE strSQL index, "True,False"
MSE.default index, mycstr(objRST("subtotal_flag"))

'Optional Effort ID
index = MSE.additem
MSE.fieldtype index, "TEXT"
MSE.ColumnHeader index, "OPTIONAL EFFORT FLAG"
MSE.default index, mycstr(objRST("optional_effort_id"))

'Record ID
index = MSE.AddItem
MSE.fieldType index, "RECORDID"
MSE.ColumnHeader index, " "
MSE.default index, mycstr(objRST("rec_id"))
objRST.movenext
wend ' objRST
objRST.close
Table Headers
%> <form method="POST" action="<%=l_this_file%>" name="Group_Edit_Form" cellpadding="3" cellspacing="3">
<table>

<%
'response.write (Session("RECORDIDLIST") & " Hello World")

'Set record count
%> <input type="HIDDEN" value="<%=RecsPerPage%>" name="p_RecCount"> <%
'Display Table
MSE.DisplayEditTable

```



## Integrated Marketing Technology

Page 174 of 200

ms\_editpanel.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/navigation.asp" -->
<!-- #include file="../support/objedit.asp" -->
<%
l_this_file = "ms_editpanel.asp"
Session("refer_page") = "ms_editpanel.asp"
if (Request.Form("setupTab") < "0" and Request.Form("setupTab") > " ") then Response.Redirect(Request.Form("setupTab"))

savechanges = False
page = request.querystring("pg")
if not len(page) > 0 then
    page = request.form("pg")
    savechanges = True
end if

'Update
if savechanges = True then

    'Get records
    recordcount = Request.form("p_RecCount")
    recordid = ""
    recordidlist = ""
    'response.write (" RecordCount -> " & cstr(recordcount) & "<br>")

    'response.write ("There are " & cstr(recordcount) & " records <br>")
    for i = 1 to recordcount
        recordid = Request.form("rec_id" & cstr(i))
        if recordid > " " then

            'init obj
            set PGE = new objPostGroupEdit
            PGE.init
            PGE.MSE_Row = i
            PGE.AddRecord recordid
            'response.write(cstr(recordid) + "<br>")

            'Set Stored Procedure info
            PGE.ClassType = "Multi Edit"
            PGE.SP = "sp_edit_panels"
            PGE.TableName = "panels"
            PGE.IsDelete = 0
            PGE.AddParam "panel_name"
            PGE.AddParam "panel_date_start"
            PGE.AddParam "panel_type"
            PGE.AddParam "panel_subtype"
            PGE.AddParam "test_type"
            PGE.AddParam "panel_pkg"
            PGE.AddParam "e_set_name"
            PGE.AddParam "r_set_name"
            'PGE.AddParam "newstand_sale_marker"
            PGE.AddParam "login"
            PGE.AddParam "rec_id"
            PGE.AddParam "IsDelete"
            objCMD.ActiveConnection = strSQLOLEDB
            objCMD.CommandText = "sp_edit_panel"
            objCMD.CommandType = adCmdStoredProc
            objCMD.parameters.refresh
            PGE.PostGroupEditsBySP

        end if
    next i
end if
end
```



```
'Multi Select
dim MSE                                     ' Multi Select object
dim strSQL                                 ' SQL Query text
dim NumOfRecords ' Total records matching whereclause - used for Prev and Next buttons
dim RecsPerPage ' max number of records to display on each page

'init obj
set MSE = new objGroupEdit
MSE.init
MSE.ClassType = "MultiSelect"
MSE.action = 1_this_file
MSE.Return_Page = "admin6a.asp"
MSE.TableHeader = "Multi Edit"
MSE.show = true
recordidlist = Session("RECORDIDLIST")
MSE.MS_Page = page

RecsPerPage = 5
MSE.MS_RecsPerPage = RecsPerPage
MSE.MS_Table = "panels"
MS_WhereClause = " where rec_id in (" & Session("RECORDIDLIST") & ")"
MS_OrderBy = " Order by SOURCE_NAME "
'response.write MS_WhereClause

'open recordset for current page
objCMD.ActiveConnection = strSQLOLEDB
objCMD.CommandText = "PagedQuery"
objCMD.CommandType = adCmdStoredProc
objCMD.parameters.refresh
objCMD.Parameters("@Page").value = page
objCMD.Parameters("@RecsPerPage") = RecsPerPage
objCMD.Parameters("@QueryTable") = "panels"
objCMD.Parameters("@WhereClause") = MS_WhereClause
objCMD.Parameters("@OrderBy") = MS_OrderBy
set objRST = objCMD.execute()
'response.write("Record ID list > " & Session("RECORDIDLIST"))
'response.write ("Page ->" + cstr(Request.querystring("pg")))
'strSQL = "pagedquery " + cstr(page) + ",5,'panels'," + chr(34) + MS_WhereClause + Chr(34) + "," + chr(34) + MS_OrderBy +
chr(34)
'response.write (strSQL)
'objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic

NumOfRecords = 0
if not objRST.eof then
    NumOfRecords = objRST("MoreRecords")
    MSE.MS_NumOfRecords = NumOfRecords
'response.write (NumOfRecords)
else
    'There are no more records
'response.write "There are no more records to edit"
response.redirect "admin6a.asp"
end if

while not objRST.eof
    MSE.NextRow

    'Panel Name
    index = MSE.AddItem
    MSE.fieldType index, "TEXT"
    MSE.ColumnHeader index, "PANEL NAME"
    MSE.default index, mycstr(objRST("panel_name"))

    'Panel Date Start
    index = MSE.AddItem
    MSE.fieldType index, "TEXT"
    MSE.ColumnHeader index, "PANEL DATE START"
    MSE.default index, mycstr(objRST("panel_date_start"))
```

```

'Panel Type
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "PANEL TYPE"
MSE strSQL index, "SELECT panel_type_name FROM corp_panel_type ORDER BY panel_type_name"
MSE.default index, mycstr(objRST("panel_type"))

'Panel SubType
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "SUBTYPE"
MSE strSQL index, "SELECT panel_sub_type_name from corp_panel_sub_type order by panel_sub_type_name"
MSE.default index, mycstr(objRST("panel_subtype"))

'Test type
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "TEST TYPE"
MSE strSQL index, "SELECT panel_test_type_name from corp_panel_test_type order by panel_test_type_name"
MSE.default index, mycstr(objRST("test_type"))

'Package Type
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "PACKAGE"
MSE strSQL index, "SELECT package_name from prod_panel_package where product_id = " &
Session("PRODUCT_ID") & " order by package_name"
MSE.default index, mycstr(objRST("panel_pkg"))

'Expense Set
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "EXPENSE SET"
MSE strSQL index, "SELECT e_set_name from expense_sets where product_id = " & Session("PRODUCT_ID") & "
order by e_set_name"
MSE.default index, mycstr(objRST("e_set_name"))

'Revenue Set
index = MSE.AddItem
MSE.fieldType index, "SELECT"
MSE.ColumnHeader index, "REVENUE SET"
MSE strSQL index, "SELECT r_set_name from revenue_sets where product_id = " & Session("PRODUCT_ID") & "
order by r_set_name"
MSE.default index, mycstr(objRST("r_set_name"))

'Newstand Marker
index = MSE.AddItem
MSE.fieldType index, "SELECTLOV"
MSE.ColumnHeader index, "NEWSTAND MARKER"
MSE strSQL index, "N,Y"
MSE.default index, mycstr(objRST("newstand_sale_marker"))

'Record ID
index = MSE.AddItem
MSE.fieldType index, "RECORDID"
MSE.ColumnHeader index, " "
MSE strSQL index, "N,Y"
MSE.default index, mycstr(objRST("rec_id"))
objRST.movenext
wend ' objRST
objRST.close
Table Headers
%>      <form method="POST" action="<%=l_this_file%>" name="Group_Edit_Form" cellpadding="3" cellspacing="3">
        <table>

<%
'response.write (Session("RECORDIDLIST") & " Hello World")

```



'Set record count

%> <input type="HIDDEN" value="<%=RecsPerPage%>" name="p\_RecCount"> <%

'Display Table

MSE.DisplayEditTable

%>





noadminrights.asp

```
<%@ Language=VBScript %>

<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoutils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<%

l_this_file = "NoAdminRights.asp"

%>

<html>
<head>
<title>You Do Not Have Admin Rights for this product!</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366"><font size="20">
<br>
You Do Not Have Admin Rights for <%=Session("Title_name")%>!!
</font>
</body>
</html>
```



nosetuprights.asp

```
<%@ Language=VBScript %>

<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoutils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<%

l_this_file = "NoSetupRights.asp"

%>

<html>
<head>
<title>You Do Not Have Setup Table Access Rights for this product!</title>

</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366"><font size="20">
<br>
You Do Not Have Setup Table Access Rights for <%=Session("Title_name")%>!!
</font>
</body>
</html>
```

searchbounce.asp

```

<%@ Language=VBScript %>
    <!-- #include file="./support/global_constants.asp" -->
    <!-- #include file="./support/dbConnection.asp" -->
    <!-- #include file="./support/adoUtils.asp" -->
    <!-- #include file="./support/srcTab.asp" -->

<%
'parameters should be:
' p_option,p_grp_name, p_PRODUCT_ID, p_SRC_CAT_NAME,
' p_SOURCE_NAME,p_SRC_INDICATOR, p_camp_name
' which url we redirect to ...
' parameters: Z => record id
' Y => option
' Response.write("This is so gay")
p_action = cint(Request.QueryString("X"))
p_option = cint(Request.QueryString("Y"))
' Response.write("<br> " & p_option)
p_rec_id = cint(Request.QueryString("Z"))
' Response.write("<br> " & p_rec_id)
' Response.Write p_grp_name

if p_option=0 then
strSQL = "select GRP_NAME, TITLE_NAME, PRODUCT_ID, SRC_CAT_NAME "
    & " from SRC_CATEGORY "
    & " where rec_id =" & p_rec_id
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
session("GRP_NAME") = objRST.Fields("GRP_NAME").Value
session("TITLE_NAME") = objRST.Fields("TITLE_NAME").Value
session("PRODUCT_ID") = objRST.Fields("PRODUCT_ID").Value
session("SRC_CAT_NAME") = objRST.Fields("SRC_CAT_NAME").Value
objRST.close

elseif p_option=1 then
strSQL = "select GRP_NAME, Src_indicator, TITLE_NAME, PRODUCT_ID, SRC_CAT_NAME, SOURCE_NAME "
    & " from SOURCE "
    & " where rec_id =" & p_rec_id
' Response.Write strSQL
objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
session("GRP_NAME") = objRST.Fields("GRP_NAME").Value
session("PRODUCT_ID") = objRST.Fields("PRODUCT_ID").Value
session("TITLE_NAME") = objRST.Fields("TITLE_NAME").Value
session("SRC_CAT_NAME") = objRST.Fields("SRC_CAT_NAME").Value
session("SOURCE_NAME") = objRST.Fields("SOURCE_NAME").Value
Session("SRC_INDICATOR") = objRST.Fields("SRC_INDICATOR").Value
objRST.close
if p_action = 1 then
    Response.Redirect("editSource.asp?Z=" & p_rec_id)
else
    Response.Redirect("admin5a.asp")
end if
elseif p_option=2 then
strSQL = "select Src_indicator, GRP_NAME, TITLE_NAME, PRODUCT_ID, SRC_CAT_NAME, SOURCE_NAME,
CAMP_NAME "
    & " from CAMPAIGNS "
    & " where rec_id =" & p_rec_id

objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
session("GRP_NAME") = objRST.Fields("GRP_NAME").Value
session("PRODUCT_ID") = objRST.Fields("PRODUCT_ID").Value
session("TITLE_NAME") = objRST.Fields("TITLE_NAME").Value
session("SRC_CAT_NAME") = objRST.Fields("SRC_CAT_NAME").Value
session("SOURCE_NAME") = objRST.Fields("SOURCE_NAME").Value
Session("CAMP_NAME") = objRST.Fields("CAMP_NAME").Value
Session("SRC_INDICATOR") = objRST.Fields("SRC_INDICATOR").Value

```



```
objRST.close
if p_action = 1 then
    Response.Redirect("editCamp.asp?Z=" & p_rec_id)
else
    Response.Redirect("admin6a.asp")
end if
elseif p_option=3 then
    strSQL = "select Src_indicator,GRP_NAME, TITLE_NAME, PRODUCT_ID, SRC_CAT_NAME, SOURCE_NAME,
CAMP_NAME, PANEL_NAME " _
    & " from PANELS " _
    & " where rec_id =" & p_rec_id

objRST.open strSQL, strSQLOLEDB, adOpenStatic, adLockOptimistic
session("GRP_NAME") = objRST.Fields("GRP_NAME").Value
session("PRODUCT_ID") = objRST.Fields("PRODUCT_ID").Value
session("TITLE_NAME") = objRST.Fields("TITLE_NAME").Value
session("SRC_CAT_NAME") = objRST.Fields("SRC_CAT_NAME").Value
session("SOURCE_NAME") = objRST.Fields("SOURCE_NAME").Value
Session("CAMP_NAME") = objRST.Fields("CAMP_NAME").Value
Session("PANEL_NAME") = objRST.Fields("PANEL_NAME").Value
Session("SRC_INDICATOR") = objRST.Fields("SRC_INDICATOR").Value
objRST.close
if p_action = 1 then
    Response.Redirect("editPanel.asp?Z=" & p_rec_id)
else
    Response.Redirect("adminkeys.htm")
end if
else
    Response.Write "<br> else this"
end if

%>
```

setupcampstat.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->
<%
    l_this_file = "setupCampStat.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "Campaign Status"
    l_descr = "The status of a Campaign <BR> (i.e. active, complete)." & vbcr

%>
<html>
<head>
<title><%Response.Write l_name%></title>

</head>

<body bgcolor="99CCFF" background="..\graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellpadding="0" cellspacing="10" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions l_Name, l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
    <% ' call setupTab's show
        dim l_titles(2)
        l_titles(0) = "Campaign Status"
        l_titles(1) = "Description"
        l_editProc = "editCampStat.asp"
        l_addProc = "addCampStat.asp"
        l_columnCount = 2

        myQ = "select camp_status_name, CAMP_STATUS_DESCR, REC_ID " _
            & " from prod_CAMP_status where " _
            & " Product_id =" & Session("PRODUCT_ID") _
            & " Order by camp_status_name "

        showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
    </td>
</tr>
</table>
</form>
</body>
</html>
```

setupcamptype.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->
<%

    l_this_file = "setupCampType.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "Campaign Type"
    l_descr = "A broad classification of a Campaign which is " _
        & " user defined and can differ between Sources. (i.e. Winter DM, hotline). " & vbcr
    %>
<html>
<head>
<title><%Response.Write l_name%></title>
</head>

<body bgcolor="#99CCFF" background="../graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions l_name, l_descr, l_showPid, l_this_file
    %>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
<% ' call setupTab's show
    dim l_titles(2)
    l_titles(0) = "Campaign Type"
    l_titles(1) = "Description"
    l_editProc = "editCampType.asp"
    l_addProc = "addCampType.asp"
    l_columnCount = 2

    myQ = "select camp_type_name, CAMP_TYPE_DESCR, REC_ID " _
        & " from PROD_CAMP_type where " _
        & " Product_id =" & Session("PRODUCT_ID") _
        & " Order by camp_type_name "

    showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
</td>
</tr>
</table>
</form>
</body>
</html>
```

setupeset.asp

```

<%@ Language=VBScript %>
<!-- #include file=../support/global_constants.asp -->
<!-- #include file=../support/dbConnection.asp -->
<!-- #include file=../support/adoUtils.asp -->
<!-- #include file=../support/srcTab.asp -->
<!-- #include file=../support/setupTab.asp -->
<!-- #include file=../support/JavaScripts.asp -->
<%
    l_this_file = "setupESet.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file)
    end if

    l_name = "Expense Set"
    l_descr = "A set of expenses which can be applied to a Campaign, Panel or Key." & vbcr

%>
<html>
<head>
<title><%Response.Write l_name%></title>
</head>

<body bgcolor="99CCFF" background=../graphics/adminhorzslice2.gif text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name=form1 id=form1>
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions l_name, l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
<% ' call setupTab's show
    dim l_titles(1)
    l_titles(0) = "Expense Set"

    l_editProc = "editESet.asp"
    l_addProc = "addESet.asp"
    l_columnCount = 1

    myQ = "select e_set_name , REC_ID " _
        & "from expense_sets " _
        & " where product_id = " & Session("PRODUCT_ID") _
        & " Order by e_set_name "

    showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
</td>
</tr>
</table>
</form>
</body>
</html>

```

setuplistcat.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->
<%
    l_this_file = "setupListCat.asp"
    l_showPid = true
    if not (session("Product_id") > " ") then
        Response.Redirect("admin2a.asp?FR=" & l_this_file)
    end if

    l_name = "List Category"
    l_descr = "A broad category of related lists." & vbcr

%>
<html>
<head>
<title><%Response.Write l_name%></title>
</head>

<body bgcolor="99CCFF" background="../graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellpadding="10" cellspacing="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions l_name, l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
    <% ' call setupTab's show
        dim l_titles(1)
        l_titles(0) = "List Category"
        l_titles(1) = "Description"
        l_editProc = "editListCat.asp"
        l_addProc = "addListCat.asp"
        l_columnCount = 2

        myQ = "select list_cat_name,list_cat_descr, REC_ID " _
            & " from list_Cats " _
            & " where product_id = " & Session("PRODUCT_ID") _
            & " Order by list_cat_name "
        showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount
    %>
</td>
</tr>
</table>
</form>
</body>
</html>

```



setuplistname.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->
<%
    l_this_file = "setupListName.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "List Name"
    l_descr = "The name of a list." & vbcr

    %>
<html>
<head>
<title><%Response.Write l_name%></title>
</head>

<body bgcolor="99CCFF" background="../graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellpadding="0" cellspacing="10" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions l_name, l_descr, l_showPid, l_this_file
    %>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
    <% ' call setupTab's show
        dim l_titles(1)
        l_titles(0) = "List Name"
        l_titles(1) = "Description"
        l_editProc = "editListName.asp"
        l_addProc = "addListName.asp"
        l_columnCount = 2

        myQ = "select list_name, list_name_description , REC_ID " _
            & " from list_names " _
            & " where product_id = " & Session("PRODUCT_ID") _
            & " Order by list_name "

        showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>

</td>
</tr>
</table>
</form>
</body>
</html>

```

setuplistsegment.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->
<%
    l_this_file = "setupListSegment.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "List Segment"
    l_descr = "A specific selection from a list. (i.e. 90 day purchasers, expires, male). " & vbcr

%>
<html>
<head>
<title><%Response.Write l_name%></title>
</head>

<body bgcolor="99CCFF" background="..\graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellpadding="10" cellspacing="10" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions "List Segment", l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
    <% ' call setupTab's show
    dim l_titles(1)
    l_titles(0) = "List Segment"
    l_titles(1) = "Description"
    l_editProc = "editListSegment.asp"
    l_addProc = "addListSegment.asp"
    l_columnCount = 2

    myQ = "select list_segment_name, list_segment_description , REC_ID " _
        & " from list_segments " _
        & " where product_id = " & Session("PRODUCT_ID") _
        & " Order by list_segment_name "

    showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
</td>
</tr>
</table>
</form>
</body>
</html>

```

setuppainelpackage.asp

```

<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->
<%
    l_this_file = "setupPanelPackage.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "Panel Package"
    l_descr = "The code or description of a specific mailed package." & vbcr

%>
<html>
<head>
<title>Panel Package</title>

</head>

<body bgcolor="99CCFF" background="..\graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name=
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption
    LeftCaptions l_name, l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
    <% ' call setupTab's show
        dim l_titles(1)
        l_titles(0) = "Panel Package"
        l_titles(1) = "Description"
        l_editProc = "editPanelPackage.asp"
        l_addProc = "addPanelPackage.asp"
        l_columnCount = 2

        myQ = "select Package_name, Package_description, REC_ID "
        & " from prod_panel_package where product_id = " & Session("PRODUCT_ID") _
        & " Order by Package_name "

        showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
    </td>
</tr>
</table>
</form>
</body>
</html>

```

setuppanelsubtype.asp

```

<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->

<html>
<head>
<title>Panel Sub Type</title>

</head>

<body bgcolor="#99CCFF" background="../graphics/adminhorzslice2.gif" text="#000000"
      link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption
l_descr = "Another field used to further define a set of related Panels (if necessary). (i.e. sweepstakes, newstand, inserts)." & vbcr

l_showPid = false

      LeftCaptions "Panel Sub Type", l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
      <% ' call setupTab's show
      dim l_titles(1)
      l_titles(0) = "Panel Sub Type"
      l_titles(1) = "Description"

      l_editProc = "editPanelSubType.asp"

      l_addProc = "addPanelSubType.asp"
      l_columnCOUNt = 2

      myQ = "select panel_sub_type_name,panel_sub_type_descr, REC_ID " _
      & " from corp_panel_sub_type " _
      & " Order by panel_sub_type_name "

      showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCOUNt %>
</td>
</tr>
</table>
</form>
</body>
</html>

```

setuppaneltesttype.asp

```
<%@ Language=VBScript %>
<!-- #include file="./support/global_constants.asp" -->
<!-- #include file="./support/dbConnection.asp" -->
<!-- #include file="./support/adoUtils.asp" -->
<!-- #include file="./support/srcTab.asp" -->
<!-- #include file="./support/setupTab.asp" -->
<!-- #include file="./support/JavaScripts.asp" -->

<html>
<head>
<title>Panel Sub Type</title>

</head>

<body bgcolor="99CCFF" background="./graphics/adminhorzslice2.gif" text="#000000"
      link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellpadding="10" cellspacing="10" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption
l_descr = "A field used to define a set of related panels by test method (i.e. price, offer, term). " & vbcr
l_showPid = false

      LeftCaptions "Panel Test Type", l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
      <% ' call setupTab's show
      dim l_titles(1)
      l_titles(0) = "Panel Test Type"
      l_titles(1) = "Description"

      l_editProc = "editPanelTestType.asp"
      l_addProc = "addPanelTestType.asp"
      l_columnCount = 2

      myQ = "select panel_test_type_name,panel_test_type_descr, REC_ID " _
      & " from corp_panel_test_type " _
      & " Order by panel_test_type_name "

      showSetupTable l_titles, myQ, l_editProc,l_addProc, l_columnCount %>
</td>
</tr>
</table>
</form>
</body>
</html>
```

setuppaneltype.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->
<html>
<head>
<title>Panel Type</title>

</head>

<body bgcolor="#99CCFF" background="..\graphics/adminhorzslice2.gif" text="#000000"
link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name>
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption
l_descr = "A set of related panels typically used to distinguish control panels from test panels. " & vbcr

l_showPid = false

LeftCaptions "Panel Type", l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
<% ' call setupTab's show
dim l_titles(1)
l_titles(0) = "Panel Type"
l_titles(1) = "Description"
l_editProc = "editPanelType.asp"
l_addProc = "addPanelType.asp"
l_columnCount = 2

myQ = "select panel_type_name,panel_type_descr, REC_ID " _
& " from corp_panel_type " _
& " Order by panel_type_name "

showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>

</td>
</tr>
</table>
</form>
</body>
</html>
```

setuppriorsrcdesc.asp

```
<%@ Language=VBScript %>
<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/setupTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->
<%
    l_this_file = "setupPriorSrcDesc.asp"

    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "Source Identifier"
    l_desc = "Describes Prior Sources" & vbcr
%>

<html>
<head>
<title>Prior Source Description</title>

</head>

<body bgcolor="#99CCFF" background="..\graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action="form1" id="form1">
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

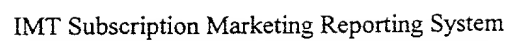
<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

    LeftCaptions l_name, l_desc, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
<% ' call setupTab's show
    dim l_titles(6)
    l_titles(0) = "Source Identifier"
    l_titles(1) = "Renewal Prior Source Description"
    l_titles(2) = "Billings Current Source Description"
                l_titles(3) = "Model Source Category"
                l_titles(4) = "Model Source Description"
                l_titles(5) = "Agent Source Description"
                l_editProc = "editPriorSrcDesc.asp"

    l_addProc = "addPriorSrcDesc.asp"
    l_columnCount = 6

    myQ = "SELECT prior_source_identifier, prior_source_desc, billing_cur_src_desc, model_source_cat, model_source,
agent_src_desc, REC_ID " _
        & " FROM prior_source_desc WHERE " _
        & " Product_id =" & Session("PRODUCT_ID") _
        & " ORDER BY prior_source_identifier "

    showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
</td>
</tr>
</table>
</form>
</body>
</html>
```





setuprset.asp

```
<%@ Language=VBScript %>
<!-- #include file="./support/global_constants.asp" -->
<!-- #include file="./support/dbConnection.asp" -->
<!-- #include file="./support/adoUtils.asp" -->
<!-- #include file="./support/srcTab.asp" -->
<!-- #include file="./support/setupTab.asp" -->
<!-- #include file="./support/JavaScripts.asp" -->
<%
    l_this_file = "setupRSet.asp"
    l_showPid = true
    if not ( session("Product_id") > " " ) then
        Response.Redirect("admin2a.asp?FR=" & l_this_file )
    end if

    l_name = "Revenue Set"
    l_descr = "Revenue over and above subscription revenue, applied to a Campaign, Panel or Key. (i.e. List Rental Revenue, Advertising Revenue). "

%>
<html>
<head>
<title><%Response.Write l_name%></title>
</head>

<body bgcolor="#99CCFF" background="./graphics/adminhorzslice2.gif" text="#000000"
    link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name=form1 id=form1>
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption

        LeftCaptions l_name, l_descr, l_showPid, l_this_file
    %>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
        <% ' call setupTab's show
            dim l_titles(1)
            l_titles(0) = "Revenue Set"

            l_editProc = "editRSet.asp"
            l_addProc = "addRSet.asp"
            l_columnCount = 1

            myQ = "select r_set_name , REC_ID " _
                & "from revenue_sets " _
                & " where product_id = " & Session("PRODUCT_ID") _
                & " Order by r_set_name "

            showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
        </td>
</tr>
</table>
</form>
</body>
</html>
```

setupsourcename.asp

```
<%@ Language=VBScript %>
<!-- #include file="../support/global_constants.asp" -->
<!-- #include file="../support/dbConnection.asp" -->
<!-- #include file="../support/adoUtils.asp" -->
<!-- #include file="../support/srcTab.asp" -->
<!-- #include file="../support/setupTab.asp" -->
<!-- #include file="../support/JavaScripts.asp" -->

<html>
<head>
<title>Source Description</title>

</head>

<body bgcolor="#99CCFF" background="../graphics/adminhorzslice2.gif" text="#000000"
      link="#000000" vlink="#000000" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">
<form method="post" action name=form1 id=form1>
<table width="660" border="0" cellspacing="10" cellpadding="0" valign="top">

<tr valign="top">
<td valign="top">
<% ' call setupTab's Left Caption
l_descr = "Describes a specific channel of sale." & vbcr
l_showPid = false

      LeftCaptions "Source Description", l_descr, l_showPid, l_this_file
%>
</td>
<td rowspan="6" width="468" valign="top" ALIGN="LEFT">
      <% ' call setupTab's show
      dim l_titles(1)
      l_titles(0) = "Source"
      l_titles(1) = "Description"
      l_editProc = "editSourceName.asp"
      l_addProc = "addSourceName.asp"
      l_columnCount = 2

      myQ = "select u_source_name, u_source_description, REC_ID " _
      & " from Source_descriptions " _
      & " Order by u_source_name "

      showSetupTable l_titles, myQ, l_editProc, l_addProc, l_columnCount %>
</td>
</tr>
</table>
</form>
</body>
</html>
```



setuptabbounce.asp

```
<%@ Language=VBScript %>  
<% Response.Redirect(Request.Form("setupTab")) %>
```

tabautosetup.asp

```
<%@ Language=VBScript %>

<!-- #include file="..\support/global_constants.asp" -->
<!-- #include file="..\support/dbConnection.asp" -->
<!-- #include file="..\support/adoUtils.asp" -->
<!-- #include file="..\support/srcTab.asp" -->
<!-- #include file="..\support/JavaScripts.asp" -->

<%

l_this_file = "tabautosetup.asp"
l_showPid = true
if not ( session("Product_id") > " " ) then
    Response.Redirect("admin2a.asp?FR=" & l_this_file )
end if

If Request.form("RUNAUTOSETUP") <> "" then 'run autosetup_wizard stored proc
    'response.write "<font size=5>Executing AutoSetup Wizard ...</font><p>"
    objCMD.ActiveConnection = strSQLOLEDB
    objCMD.CommandText = "autosetup_wizard"
    objCMD.CommandType = adCmdStoredProc
    objCMD.Parameters.refresh
    objCMD.Parameters(1).Value = cint(0) '0 for no returned text,
non-zero to print output
    objCMD.Parameters(2).value = cint(Session("Product_id")) 'product_id

    Set TempNonSessionVar = objCMD.Execute()
    'response.redirect("tabautosetup.asp")
end if

l_this_file = "admin4a.asp"

Session("refer_page") = "admin4a.asp"
Session("refer_Title") = "Source"

%>
<head><title>tabautosetup.asp</title>
<% LorieScripts %>
</head>

<body bgcolor="#99CCFF" vlink="000000" link="003366"
onLoad="MM_preloadImages('graphics/editablero1222.gif',#945899226033');MM_preloadImages('graphics/editablero1222.gif',#945899269583')">
<% <body bgcolor="#99CCFF" vlink="000000" link="003366"> %>

<table width="640" border="0" cellpadding="2">
<% prevSelections (3) %>

    <form method="post" action="tabautosetup.asp" id="form2" name="form2">
<tr valign="BOTTOM">
    <td width="65">

        </td>
        <td width="500">
            <p> <%
strSQL = "select source_name, src_indicator, valid_from_date, valid_to_date, setup_mask, e_set_name, rec_id " & _
        " from autosetup_scheme " & _
        " where product_id = " & Session("PRODUCT_ID") & _
        " order by src_cat_name "
        %>
            </td>
        </tr>
    </form>
</tr>
<tr valign="top">
    <td width="640" colspan="3">
        <p>&nbsp;</p>
    </td>
</tr>
```

```

</td>
</tr>
</table>
<%

```

```
l_query = strSQL
```

```

l_columnCount = 6
l_addProc = "add_autosetup.asp"
l_editProc = "edit_autosetup.asp"
dim l_title(6), l_columnWidth(6)

```

```

l_title(0) = " Source "
l_title(1) = " Source Indicator "
l_title(2) = " Select Date From "
l_title(3) = " Select Date To "
l_title(4) = " Setup Mask "
l_title(5) = " Expense Set "
l_columnWidth(0) = "166"
l_columnWidth(1) = "41"
l_columnWidth(2) = "50"
l_columnWidth(3) = "50"
l_columnWidth(4) = "50"
l_columnWidth(5) = "50"

```

```
%>
```

```

<form method=post action="tabautosetup.asp" id="formid" name="formname">
<a href="http://www.imtnetwork.com/help/auto_setup_help/index.html" target="_blank">AutoSetup

```

```
Instructions</a><br><br>
```

```

<input type="SUBMIT" Value="Run Auto-Setup Wizard" Name="RUNAUTOSSETUP" id="RUNAUTOSSETUP">
</form>

```

```
<font size="5">AutoSetup Schemes</font><p>
```

```
<% adminTable l_Title, l_Query, l_editProc, l_addProc, l_columnWidth, L_ColumnCount
```

```
l_query = "select src_cat_name, campaign_code, campaign_description, campaign_date, panel_scheme, rec_id
```

```
" & _
```

```

" from autosetup_campaign_def " & _
" where product_id = " & Session("product_id") & _
" order by src_cat_name, campaign_description, campaign_code "

```

```
'redim l_title(4)
```

```
'redim l_columnWidth(4)
```

```
l_columnCount = 5
```

```
l_addProc = "add_campaign_def.asp"
```

```
l_editProc = "edit_campaign_def.asp"
```

```
l_title(0) = " Source Category "
```

```
l_title(1) = " Campaign Code "
```

```
l_title(2) = " Campaign Description "
```

```
l_title(3) = " Campaign Date "
```

```
l_title(4) = " Panel Scheme "
```

```
l_columnWidth(0) = "50"
```

```
l_columnWidth(1) = "50"
```

```
l_columnWidth(2) = "50"
```

```
l_columnWidth(3) = "50"
```

```
l_columnWidth(4) = "50"
```

```
%>
```

```
<p>
```

```
<p>
```

```
<font size="5">Campaign Name Definitions</font><p>
```

```
<% adminTable l_Title, l_Query, l_editProc, l_addProc, l_columnWidth, L_ColumnCount
```

```
l_query = "select source_name, source_indicator, prior_source_indicator, valid_from_date, valid_to_date,
```

```
rec_id " & _
```

```

" from autosetup_renewal_psdef " & _
" where product_id = " & Session("product_id") & _
" order by src_cat_name "

```





## IMT REPORTING SYSTEM STORED PROCEDURES (USER)

STORED PROCEDURE
Add keys
Addrptdef
Autosetup_wizard
Cascade update
Concat c codes
Concat p codes
Create r101 topline
Del key
Do101 flow seldatfix
Doavail401keys
Doavail401keysbypid
Doavailkeys
Doavailkeysbypid
Dobaserankinfo
Docampe sets
Docamprsets
Doestmaxeffort
Domw
Dopanelesets
Dopanelrsets
Doprodsetup
Dor101 summary
Dor401 summary
Dorecyclekey
Dorecyclepanelkey
Dosourceinds
Dosummary
Dosummarydb
Expenses
Getcorpinfo
Maxbilleffort
Maxeffort
Maxgifteffort
Pagedquery
Rebuild rpt cats
Show columns
Sp add autosetup schemes
Sp add campaign
Sp add campaign def
Sp add corp panel sub type
Sp add corp panel test type
Sp add corp panel type
Sp add keys
Sp add keys2
Sp add panel
Sp add panel def
Sp add prior source desc
Sp add prod camp status



STORED PROCEDURE
Sp add prod camp type
Sp add prod e set
Sp add prod list cat
Sp add prod list name
Sp add prod list segment
Sp add prod panel package
Sp prod r set
Sp add renewal psdef
Sp add source
Sp add source description
Sp cascade product id
Sp deletecamp statuses
Sp deletecamp types
Sp deletepanel subtypes
Sp deletepanel testtypes
Sp deletepanel types
Sp deleterpt def
Sp edit autosetup
Sp edit autosetup campaign def
Sp edit autosetup panel def
Sp edit autosetup renewal psdef
Sp edit campaign
Sp edit corp panel sub type
Sp edit corp panel test type
Sp edit corp panel type
Sp edit key
Sp edit panel
Sp edit prior source desc
Sp edit prod camp status
Sp edit prod camp type
Sp edit prod e set
Sp edit prod list cat
Sp edit prod list name
Sp edit prod list segment
Sp edit prod panel package
Sp edit prod r set
Sp edit source
Sp edit source description
Sp loginfo
Sp deletecamp types
Sp deletepanel subtypes
Sp deletepanel testtypes
Sp deletepanel types
Sp deleterpt def
Sp edit autosetup
Sp edit autosetup campaign def
Sp edit autosetup panel def
Sp edit autosetup renewal psdef
Sp edit campaign
Sp edit corp panel sub type
Sp edit corp panel test type
Sp edit corp panel type





STORED PROCEDURE
Sp edit key
Sp edit panel
Sp edit prior source desc
Sp edit prod camp status
Sp edit prod camp type
Sp edit prod e set
Sp edit prod list cat
Sp edit prod list name
Sp edit prod list segment
Sp edit prod panel package
Sp edit prod r set
Sp edit edit source
Sp edit source description
Sp loginfo
Sp removeduplicate101keys
Sp removeduplicate401keys
Sp updatecamp statuses
Sp updatecamp types
Sp updatepanel subtypes
Sp updatepanel testtypes
Sp updatepanel type
Sp getrptlist



## Procedure Name: Add\_keys

```
/****** Object: Stored Procedure dbo.add_Keys Script Date: 1/14/99 3:43:07 PM *****/
/****** Object: Stored Procedure dbo.add_Keys Script Date: 1/12/99 11:05:37 PM *****/
/****** Object: Stored Procedure dbo.add_Keys Script Date: 11/28/98 2:31:13 PM *****/
/****** Object: Stored Procedure dbo.add_Keys Script Date: 11/17/98 2:43:36 PM *****/
/****** Object: Stored Procedure dbo.add_Keys Script Date: 10/19/98 6:03:58 PM *****/
/****** Object: Stored Procedure dbo.add_Keys Script Date: 10/6/98 7:56:48 PM *****/
/****** Object: Stored Procedure dbo.add_Keys Script Date: 10/6/98 11:08:02 AM *****/
```

```
@full_promo_key varchar (25),
@dt_valid_from smalldatetime,
@dt_valid_to smalldatetime,
@src_cat_name varchar(30),
@src_indicator char (3),
@panel_name varchar (250),
@camp_name varchar (250),
@product_id numeric(8,0),
@source_name varchar(250)
```

AS

insert into keys

```
(
    full_promo_key,

    dt_valid_from,
    dt_valid_to,
    src_cat_name,
    src_indicator,
    panel_name,
    camp_name,
    product_id,
    source_name
```

)

values

```
(
    @full_promo_key,
    @dt_valid_from,
    @dt_valid_to,
    @src_cat_name,
    @src_indicator,
    @panel_name,
    @camp_name,
    @product_id,
    @source_name
```

)

Delete from avail\_keys where full\_promo\_key = @full\_promo\_key and product\_id=@product\_id



Procedure Name: Addrptdef

```
/*
    (
        @parameter1 datatype = default value,
        @parameter2 datatype OUTPUT
    )
*/
(@rpt_group varchar(75),
 @rpt_title varchar(75),
 @rpt_description varchar(250),
 @rpt_selFormula varchar(1000),
 @rpt_path varchar(250),
 @rpt_param1 varchar(50),
 @rpt_param2 varchar(50),
 @rpt_param3 varchar(50),
 @rpt_param4 varchar(50),
 @product_id varchar(250),
 @login varchar(50)
)

As
    /* set nocount on */
    Insert into rpt_Defs
    (rpt_group,rpt_title,rpt_description,rpt_selFormula,rpt_path,rpt_param1,rpt_param2,product_id)
    Values
    (@rpt_group,@rpt_title,@rpt_description,@rpt_selFormula,@rpt_path,@rpt_param1,@rpt_param2,@product_id)

execute sp_LogInfo @product_id,@login,'Saved Report',@rpt_title
```

## Procedure Name: Autsetupwizard

```
@verbose int -- 0 means no text, non-zero means display text messages
, @product_id int

as
begin --autsetup wizard
if @product_id > 0 --product id is set to -1 to run autsetup_wizard for ALL PRODUCTS. This happens is
doavail401keys
begin
    declare avail_keys cursor
    for select product_id
        , full_promo_key
        , fulfillment_house
        , dt_valid_to
        , source_identifier
        , mail_qty
    from avail_keys
    where fulfillment_house is not null
    and product_id = @product_id
    --or product_id = 360
    --or full_promo_key like "6RAA_K_529"

end
else
begin
    declare avail_keys cursor
    for select product_id
        , full_promo_key
        , fulfillment_house
        , dt_valid_to
        , source_identifier
        , mail_qty
    from avail_keys
    where fulfillment_house is not null

end

--Avail Key Vars
--declare @product_id varchar(50)
declare @avail_key varchar(50)
declare @fulfillment_house varchar(50)
declare @dt_valid_to datetime
declare @source_identifier char(1)
declare @mail_qty int

--Campaign vars
declare @campaign_code varchar(50)
declare @campaign_description varchar(50)
declare @rec_id int
declare @src_indicator char(3)
declare @camp_name varchar(250)
declare @title_name varchar(250)
declare @camp_date_fr datetime

--Setup Wizard vars
```

```

declare @setup_mask varchar(50)
declare @src_cat_name varchar(50)
declare @source_name varchar(250)
declare @prior_source_indicator varchar(2)
declare @valid_from_date datetime
declare @valid_to_date datetime
declare @grp_name varchar(50)
declare @e_set_name varchar(50)

--panel vars
declare @panel_name varchar(250)
declare @panel_code varchar(250)
declare @panel_scheme varchar(50)

--local vars
declare @i int
declare @number_of_valid_schemes int

if @verbose > 0 print "Starting ..."
open avail_keys
fetch next from avail_keys into @product_id, @avail_key, @fulfillment_house, @dt_valid_to,
@source_identifier, @mail_qty
while @@fetch_status = 0 --Loop through avail keys and attempt to assign them to campaigns
begin
    if @verbose > 0 print "Avail Key " + @avail_key
    if @verbose > 0 print "Source Identifier " + @source_identifier
    --Get autoseup vars for schemes matching current key
    set @setup_mask = null
    --If the number of valid schemes is greater than zero and the source category is renewals
    then the source is split
    --and a match is required from the prior source definition table(autoseup_renewal_psdef)
    select @number_of_valid_schemes = count(*)
        from autoseup_scheme
        where product_id = @product_id
        and valid_from_date <= @dt_valid_to
        and valid_to_date >= @dt_valid_to
        and src_indicator = @source_identifier
    ---GREGG ADDED THE FOLLOWING CONDITION TO THE VALID
    SCHEME COUNTER
    and setup_mask = @setup_mask
    select @setup_mask = setup_mask
        , @src_cat_name = src_cat_name
        , @src_indicator = src_indicator
        , @source_name = source_name
        , @valid_from_date = valid_from_date
        , @valid_to_date = valid_to_date
        , @grp_name = grp_name
        , @e_set_name = isnull(e_set_name, "Undefined")
        from autoseup_scheme
        where product_id = @product_id
        and valid_from_date <= @dt_valid_to
        and valid_to_date >= @dt_valid_to
        and src_indicator = @source_identifier
        and len(setup_mask) = len(@avail_key)

```

```

if @verbose <> 0 print "Source Indicator ==> " + @src_indicator
if @setup_mask is not null --Build campaign code based on setup mask, otherwise move
on to next scheme
begin
    set @campaign_code = null
    set @camp_name = null

    --if len(@avail_key) = 11 and @src_cat_name = "RENEWALS" set
    @setup_mask = "SxxxxZZxCCC"
    if @verbose <> 0 print "Found an applicable setup mask! ==> " + @setup_mask
    if @fulfillment_house = "CENTROBE" or @fulfillment_house = "PALM
    COAST" exec concat_c_codes @setup_mask, @avail_key, @campaign_code output
    if @fulfillment_house = "CDS" and (@src_cat_name = "DTP" or
    @src_cat_name = "GIFT") exec concat_c_codes @setup_mask, @avail_key, @campaign_code output
    if @fulfillment_house = "CDS" and (@src_cat_name = "RENEWALS" or
    @src_cat_name = "BILLING")
    begin --set code to pos2 & last pos then match value with def table and set
    description
        set @campaign_code = substring(@avail_key,2,1) +
    substring(@avail_key,len(@avail_key),1)

    end --if
    if @verbose <> 0 print "campaign code ==>" + @campaign_code

    select @camp_name = campaign_description
        , @camp_date_fr = campaign_date
        from autosetup_campaign_def
    where campaign_code = @campaign_code
    --and source_identifier = substring(@avail_key,1,1)
        and src_cat_name = @src_cat_name
        and product_id = @product_id

    if @camp_name is null --Build new campaign name
    begin
        set @camp_name = ""
        if @verbose <> 0 print "Source Category Name ==> " +
    @src_cat_name

        if @verbose <> 0 print "Fulfillment House ==> " +
    @fulfillment_house

        if @fulfillment_house = "CENTROBE"
        begin
            if @src_cat_name = "DTP" or @src_cat_name = "GIFT" set
    @camp_name = substring(@avail_key, 6, 4) + " Campaign " + @campaign_code
            --if @src_cat_name = "RENEWALS" set @camp_name =
    "Expire " + @campaign_code

            if @src_cat_name = "RENEWALS"
            begin
                set @camp_name = null
                select @camp_name = campaign_description
                    , @camp_date_fr = campaign_date

                    from autosetup_campaign_def
                where src_cat_name = @src_cat_name
                and campaign_code = @campaign_code
            end
        end
    end

```

```

                                and product_id = @product_id
                                if @camp_name is null
                                begin
                                    set @camp_name = "Expire " +
@campaign_code
                                    set @camp_date_fr = null
                                end

                                end
                                if @src_cat_name = "BILLING" set @camp_name =
substring(@avail_key,5,2) + " Credit Period " + @campaign_code
                                end
                                if @fulfillment_house = "PALM COAST"
                                begin
                                    if @src_cat_name = "DTP" or @src_cat_name = "GIFT" set
@camp_name = substring(@avail_key, len(@avail_key) - 3, 4) + " Campaign " + @campaign_code
                                    if @src_cat_name = "RENEWALS" set @camp_name =
substring(@avail_key, len(@avail_key) - 3, 4) + " Expire " + @campaign_code
                                    if @src_cat_name = "BILLING" set @camp_name = "Credit
Period " + @campaign_code
                                end
                                if @fulfillment_house = "CDS"
                                begin
                                    if @src_cat_name = "DTP" or @src_cat_name = "GIFT"
                                begin
                                    if substring(@avail_key, len(@avail_key) - 1, 2) < 8
                                begin
                                    set @camp_name = "200" +
substring(@avail_key, len(@avail_key) - 1, 2) + " Campaign " + @campaign_code
                                end
                                    else
                                begin
                                    set @camp_name = "199" +
substring(@avail_key, len(@avail_key) - 1, 2) + " Campaign " + @campaign_code
                                end
                                    --end --if
                                end
                                if @src_cat_name = "RENEWALS" set @camp_name =
@campaign_code
                                if @src_cat_name = "BILLING" set @camp_name =
@campaign_code
                                end

                                if @camp_name is null set @camp_name = "Auto setup Campaign -
Fulfillment House not Set!!!"
                                end --if
                                if @verbose > 0 print "Campaign Name ==> " + @camp_name

                                --set prior source indicator
                                set @i = 0
                                set @prior_source_indicator = ""
                                while @i <= len(@setup_mask)
                                begin
                                    set @i = @i + 1

```

```

        if substring(@setup_mask,@i,1)="Z" set @prior_source_indicator =
@prior_source_indicator + substring(@avail_key, @i,1)
    end --while
    if @verbose <> 0 print "Prior Source Indicator ==> " + @prior_source_indicator

    --set source name from renewal table
    --if the number of schemes is > 1 then the source is split - a match must occur
    if @src_cat_name = "RENEWALS" and @number_of_valid_schemes > 1 set
@source_name = null
        if @src_cat_name = "RENEWALS" select @source_name = source_name
            from autosetup_renewal_psdef
            where product_id = @product_id
            and src_cat_name = "RENEWALS"
            and patindex('%'+
substring(@avail_key,1,1) + '%', source_indicator) > 0
                and @dt_valid_to >= valid_from_date
                and @dt_valid_to <= valid_to_date
                --and source_indicator =

substring(@avail_key,1,1)
                --and prior_source_indicator =

@prior_source_indicator
                and patindex('%'+ @prior_source_indicator
+ '%', prior_source_indicator) > 0
            if @verbose <> 0 print "Source Name ==> " + @source_name
            if @source_name is null and @verbose <> 0 print "No match to prior source in
the psdef table!" --if no match was made, move to next scheme
            else
            begin
                --if a campaign exists use it, if not then create campaign
                set @rec_id = null
                select @rec_id = rec_id
                    from campaigns
                    where (campaign_code = @campaign_code or camp_name =
@camp_name)
                    and src_indicator = @src_indicator
                    and src_cat_name = @src_cat_name
                    and product_id = @product_id
                    and source_name = @source_name
                    --and camp_date_fr <= @dt_valid_to
                    --and camp_date_to >= @dt_valid_to
                if @rec_id is not null
                select @camp_name = camp_name
                    from campaigns
                    where rec_id = @rec_id
                if @rec_id is null and @verbose <> 0 print "Campaign Not Found!"
                if not (@rec_id is null) and @verbose <> 0 print "Campaign Rec ID
==> " + cast(@rec_id as varchar(50))

                select @title_name = title_name
                    from corp_info
                    where product_id = @product_id
                if @verbose <> 0 print "Title Name ==> " + @title_name

                if (@rec_id is null) --no campaign exists, create one; add row to
campaign def

```



```
begin
insert into campaigns
    select @src_indicator as src_indicator
    , @src_cat_name as src_cat_name
    , @product_id as product_id
    , "Undefined" as camp_type
    , @camp_name as camp_name
    , "Undefined" as camp_status_name
    , null as agt_code
    , null as agt_descr
    , @camp_date_fr as camp_date_fr
    , null as camp_date_to
    , @e_set_name as e_set_name
    , "Undefined" as r_set_name
    , @grp_name as grp_name
    , @title_name as title_name
    , @source_name as source_name
    , null as newstand_sale
    , null as budgeted_gross_pct
    , null as budgeted_vol
    , @campaign_code as campaign_code

set @rec_id = null
select @rec_id = rec_id
    from autosetup_campaign_def
    where product_id = @product_id
    and campaign_code = @campaign_code
    and src_cat_name = @src_cat_name
if @rec_id is null
    insert into autosetup_campaign_def
    (product_id
    , campaign_code
    , campaign_description
    , src_cat_name
    , campaign_date
    , panel_scheme)
    values (@product_id
    , @campaign_code
    , @camp_name
    , @src_cat_name
    , @camp_date_fr
    , null)
end --if
--ok, campaign is done, now for the panel
--Check for panel naming scheme
set @panel_scheme = null
set @panel_code = null
exec concat_p_codes @setup_mask, @avail_key, @panel_code output

select @panel_scheme = panel_scheme
    from autosetup_campaign_def
    where product_id = @product_id
    and campaign_code = @campaign_code
    --and campaign_description = @campaign_description
    and src_cat_name = @src_cat_name
```

```

--and campaign_date = @camp_date_fr
if @verbose <> 0 print "Panel Scheme ==> " + @panel_scheme

if @panel_scheme is not null and @panel_code is not null --use panel
scheme to determine panel name
begin
    set @panel_name = null
    select @panel_name = panel_name
        from autosetup_panel_def
        where product_id = @product_id
        and panel_scheme = @panel_scheme
        and patindex('%' + @panel_code + '%', panel_codes)
> 0

--gw print statement below
if @verbose <> 0 print "Panel Name ==> " + @panel_name
end
if @panel_scheme is null or (@panel_scheme is not null and
@panel_code is null) or (@panel_scheme is not null and @panel_name is null)--build panel name from
default
begin
    if not(@panel_code > " ") set @panel_name = "Control"
    else set @panel_name = "Panel " + @panel_code
    if @verbose <> 0 print "Panel Name ==> " + @panel_name
end -- if

--check to see if a panel exists
set @rec_id = null
select @rec_id = rec_id
    from panels
    where campaign_code = @campaign_code
    and src_indicator = @src_indicator
    and panel_code = @panel_code
    and src_cat_name = @src_cat_name
    and source_name = @source_name
    and product_id = @product_id
--check for unique panel name
if @rec_id is null select @rec_id = rec_id
    from panels
    where panel_name = @panel_name
    and camp_name = @camp_name
    and src_indicator = @src_indicator
    and source_name = @source_name
    and panel_name = @panel_name
    and src_cat_name = @src_cat_name
    and product_id = @product_id
if @verbose <> 0 Print "Panel ID (null means create a new one) ==> "
+ cast(@rec_id as varchar(50))

if (@rec_id is null) --then panel does not exist so create it
begin
    insert into panels
        select @camp_name as camp_name
        , @src_indicator as src_indicator
        , @panel_name as panel_name
        , @src_cat_name as src_cat_name
        , @product_id as product_id

```

```
, "Undefined" as panel_type
, @e_set_name as e_set_name
, "Undefined" as panel_subtype
, "Undefined" as r_set_name
, "Undefined" as test_type
, null as panel_offer
, null as panel_offer_price
, "Undefined" as panel_pkg
, null as panel_premium
, null as panel_date_start
, null as panel_date_end
, @grp_name as grp_name
, @title_name as title_name
, @source_name as source_name
, "Undefined" as camp_type
, "Undefined" as camp_status_name
, null as agt_code
, null as agt_descr
, @valid_from_date as camp_date_fr
, @valid_to_date as camp_date_to
, null as newstand_sale_marker
, null as newstand_sale
, null as budgeted_gross_pct
, null as budgeted_vol
, @campaign_code as campaign_code
, @panel_code as panel_code
, null as order_qty
, null as dist_qty
```

```
end --if
set @rec_id = null
if @verbose > 0 print "Assigning key to keys table ..."
--OK, now add the key to the keys table
if @verbose > 0 print "Expense set name > " + @e_set_name
if (@e_set_name='0') or (@e_set_name is null) set @e_set_name =
```

"Undefined"

```
insert into keys
select @avail_key as full_promo_key
, @dt_valid_to as dt_valid_to
, @panel_name as panel_name
, @camp_name as camp_name
, @dt_valid_to as dt_valid_from
, @src_indicator as src_indicator
, @src_cat_name as src_cat_name
, @product_id as product_id
, @e_set_name as e_set_name
, null as list_id
, 0 as level1_expense
, null as list_cost_rollout_cpm
, null as list_cost_actual_cpm
, null as fixed_expense
, 'True' as subtotal_flag
, 0 as level2_expense
, "Undefined" as r_set_name
, 0 as level3_expense
```

```

, 0 as level1_revenue
, 0 as level2_revenue
, 0 as level3_revenue
, null as p_key_descr
, "Undefined" as panel_type
, "Undefined" as panel_subtype
, "Undefined" as test_type
, null as panel_offer
, null as panel_offer_price
, "Undefined" as panel_pkg
, null as panel_premium
, null as panel_date_start
, null as panel_date_end
, @grp_name as grp_name
, @title_name as title_name
, @source_name as source_name
, "Undefined" as camp_type
, "Undefined" as camp_status_name
, null as agt_code
, null as agt_descr
, null as camp_date_fr
, null as camp_date_to
, @mail_qty as key_mail_qty
, null mail_qty_override
, null as merge_purge_qty
, "Mail Qty" as list_cost_basis
, 1 as list_cost_factor
, "Undefined" as list_cat_name
, "Undefined" as list_name
, "Undefined" as list_segment_name
, null as optional_effort_id
, null as newstand_sale
, "N" as newstand_sale_marker
, null as budgeted_gross_pct
, null as budgeted_vol
, getdate() as date_assigned
, @campaign_code as campaign_code
, @panel_code as panel_code
, "Auto Setup" as dbusername
, null as order_qty
, null as dist_qty

end -- if
end -- if, no matching source name
if @verbose > 0 print "Getting next avail key ..."
fetch next from avail_keys into @product_id, @avail_key, @fulfillment_house, @dt_valid_to,
@source_identifier, @mail_qty
end --while
close avail_keys
deallocate avail_keys
delete from avail_keys
where exists
(select *
from keys k
where k.full_promo_key = avail_keys.full_promo_key
and k.product_id = avail_keys.product_id)

```



```
    if @verbose < 0 print "All done!"  
end
```



Procedure Name: Cascade\_update

```
@field_name VARCHAR(100),
@old_field_value VARCHAR(200),
@new_field_value VARCHAR(200),
@level VARCHAR(30),
@product_id Numeric(9)
AS
declare @sql NVarchar(200)

--print @field_name + " " + @field_value
IF @level = "KEYS" or @level = "PANELS" or @level = "CAMPAIGNS"
begin
    set @sql = 'UPDATE keys SET ' + @field_name + ' = @new_value where ' + @field_name + ' =
@old_value '
    if @product_id <> -1
    begin
        set @sql = @SQL + ' and Product_id = ' + cast(@product_id as VARCHAR)
    end
    exec sp_executesql @Sql, N'@new_value VARCHAR(200), @old_value
VARCHAR(200)',@new_field_value,@old_field_value
END

IF @level = "PANELS" or @level = "CAMPAIGNS"
BEGIN
    set @sql = 'UPDATE panels SET ' + @field_name + ' = @new_value where ' + @field_name + ' =
@old_value '
    if @product_id <> -1
    begin
        set @sql = @SQL + ' and Product_id = ' + cast(@product_id as VARCHAR)
    end
    exec sp_executesql @Sql, N'@new_value VARCHAR(200), @old_value
VARCHAR(200)',@new_field_value,@old_field_value
END

IF @level = "CAMPAIGNS"
BEGIN
    set @sql = 'UPDATE campaigns SET ' + @field_name + ' = @new_value where ' + @field_name + ' =
@old_value '
    if @product_id <> -1
    begin
        set @sql = @SQL + ' and Product_id = ' + cast(@product_id as VARCHAR)
    end
    exec sp_executesql @Sql, N'@new_value VARCHAR(200), @old_value
VARCHAR(200)',@new_field_value,@old_field_value
END
```

**Procedure Name: Concat\_c\_codes**

```
@setup_mask varchar(50)
, @avail_key varchar(50)
, @campaign_code varchar(50) output as

--local vars
declare @i int

begin
    set @i = 0
    set @campaign_code = ""
    while @i < len(@setup_mask) --loop through each char in mask and add to camp code where
mask letter is a C
        begin
            set @i = @i + 1
            --select @i, substring(@setup_mask, @i, 1), substring(@avail_key, @i, 1)
            if substring(@setup_mask, @i, 1) = "C" set @campaign_code = @campaign_code +
substring(@avail_key, @i, 1)
        end --while
    --return @campaign_code
end --concat c codes
```



Procedure Name: concat\_p\_codes

@setup\_mask varchar(50)  
, @avail\_key varchar(50)  
, @panel\_name varchar(50) output as

--local vars  
declare @i int

begin

```
    set @i = 0
    set @panel_name = ""
    print "Entering concat p codes..."
    print "setup mask is ==> " + @setup_mask
    print "Avail key is ==> " + @avail_key
    --print "panel name is ==> " + @panel_name
    while @i < len(@setup_mask) --loop through each char in mask and add to panel name where
    mask letter is a P
    begin
        set @i = @i + 1
        --select @i, substring(@setup_mask, @i, 1), substring(@avail_key, @i, 1)
        if substring(@setup_mask, @i, 1) = "P" set @panel_name = @panel_name +
        substring(@avail_key, @i, 1)
        --print "Panel is now ==> " + @panel_name
    end --while
    print "Panel is now ==> " + @panel_name
end --concat p codes
```





## Procedure Name: Create\_r101\_topline

```
truncate table r101_topline
INSERT INTO r101_topline
Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        sum(r101.mail_qty) promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
-- keys to r101
AND     r101.product_id = est_max_effort.product_id
AND     r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND     r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
-- other stuff
AND     r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
and     keys.src_cat_name = 'Renewals'
AND     r101.effort_identifier = est_max_effort.max_effort
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name

Insert into r101_topline
-- else case ....

Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        0 promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
-- keys to r101
AND     r101.product_id = est_max_effort.product_id
AND     r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND     r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
AND     r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
-- other stuff
```

```
and keys.src_cat_name = 'Renewals'
AND r101.effort_identifier <> est_max_effort.max_effort
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name
--
INSERT INTO R101_TOPLINE
Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        sum(newstand_sale) promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
-- keys to r101
AND r101.product_id = est_max_effort.product_id
AND r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
AND r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
-- other stuff
and keys.src_cat_name <> 'Renewals'
AND keys.subtotal_flag = 'true'
and Newstand_sale_marker = 'Y'
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name
INSERT INTO R101_TOPLINE
Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        sum(newstand_sale) promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
-- keys to r101
AND r101.product_id = est_max_effort.product_id
AND r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
AND r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
```

```
-- other stuff
and keys.src_cat_name <> 'Renewals'
AND keys.subtotal_flag = 'true'
and Newstand_sale_marker = 'Y'
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name

INSERT INTO R101_TOPLINE
Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        0 promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
-- keys to r101
AND r101.product_id = est_max_effort.product_id
AND r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
AND r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
-- other stuff
and keys.src_cat_name <> 'Renewals'
AND keys.subtotal_flag = 'true'
and Newstand_sale_marker = 'N'
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name
```

```
INSERT INTO R101_TOPLINE
Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        sum(keys.mail_qty_override) promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
```

```
-- keys to r101
AND r101.product_id = est_max_effort.product_id
AND r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
AND r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
-- other stuff
and keys.src_cat_name <> 'Renewals'
AND keys.subtotal_flag = 'false'
and keys.mail_qty_override > 0
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name
```

```
INSERT INTO R101_TOPLINE
```

```
Select  keys.Product_id as Product_id,
        keys.src_cat_name as src_cat_name,
        keys.source_name as source_name,
        keys.camp_name as camp_name,
        -- summary fields
        sum(r101.gross_subs) AS gross_subs,
        sum(r101.tot_net_subs) AS tot_net_subs,
        min(keys.camp_date_fr) AS camp_date_fr,
        max(keys.budgeted_gross_pct) AS Budgeted_gross_pct,
        max(keys.Budgeted_vol) AS Budgeted_vol,
        sum(keys.key_mail_qty) promo_qty
FROM    keys keys, r101 r101, est_max_effort est_max_effort
WHERE   keys.full_promo_key = r101.r101_key
AND     keys.product_id = r101.product_id
-- keys to r101
AND r101.product_id = est_max_effort.product_id
AND r101.source_IDENTIFIER = est_max_effort.souce_IDENTIFIER
AND r101.issue_IDENTIFIER = est_max_effort.issue_IDENTIFIER
AND r101.prior_source_IDENTIFIER = est_max_effort.prior_source_IDENTIFIER
-- other stuff
and keys.src_cat_name <> 'Renewals'
AND keys.subtotal_flag = 'false'
and keys.mail_qty_override <= 0
GROUP BY keys.Product_id,
        keys.src_cat_name,
        keys.source_name,
        keys.camp_name
```



Procedure Name: del\_key

```

/***** Object: Stored Procedure dbo.del_Key  Script Date: 1/14/99 3:43:07 PM *****/
/***** Object: Stored Procedure dbo.del_Key  Script Date: 1/12/99 11:05:37 PM *****/
CREATE PROCEDURE del_Key
    @full_promo_key varchar (25),@product_id numeric(8,2)
AS
delete from keys where product_id=@product_id and full_promo_key = @full_promo_key

```



Procedure Name: do\_101flow\_selDatefix

```
@product_id numeric(9,2),      -- product_id
@Promo_Key varchar(25),        -- promo key being inserted
@reorg_date smalldatetime,     -- the reorg date from the file that the key comes from
@import_file_type char(1) -- the type of import file i.e. new business/cds , etc.

AS
declare
--execute do101_flow_selDateFix @product_id, @Promo_Key, @reorg_date
*@nPromo_Key varchar(25),      -- promo_key minus the right 6 characters + @maxSel_Date +
                                -- existing modified promo key from a prior calculation
    @ePromo_Key varchar(25),
    @Key_Length integer,
    @maxSel_Date smalldatetime -- max select date if prior same key exists
select @nPromo_Key = len(@Promo_Key) --promo_key length
If(@import_file_type='C')
    BEGIN
        declare c_flow cursor for
        select max(select_date) m, r101_key from r101_flow where
        product_id = @product_id and
        r101_key like (substring(@Promo_Key,1,@nPromo_Key-9)+'[ _G _]%' ) and
        (len(r101_key) > 17 OR IMPORT_file_type='C') group by select_date,r101_key order by m desc
        print 'searching for key match on C imp file type'
        END
    else
        BEGIN
        declare c_flow cursor for
        select max(select_date) m, r101_key from r101_flow where
        product_id = @product_id and r101_key like (substring(@Promo_Key,1,@nPromo_Key-6)+'%')
        and len(r101_key) < 18 group by select_date,r101_key order by m desc
        END
    open c_flow
    fetch c_flow into @maxSel_Date, @ePromo_Key
    --print 'fetch_status=' + convert(varchar,@@fetch_status)
    If (@@fetch_status = 0)
        BEGIN
            If (@maxSel_Date > DATEADD(day,-555,getDate()))
                BEGIN
                    print convert(varchar, @maxSel_Date) + 'fetch status = match found(0)'
                    update r101_flow set select_date = @maxSel_Date, r101_key = @ePromo_Key
                    where product_id = @product_id and r101_key=@Promo_Key
                    close c_flow
                    deallocate c_flow
                    return
                END
            ELSE -- no prior matching keys or select date is null
                BEGIN
                    print 'appending date to key'
                    select @maxSel_Date = @reorg_date
                    update r101_flow set select_date = @maxSel_Date,
                    r101_key = (substring(@Promo_Key,1,@nPromo_Key-6) + CONVERT (varchar,
                    @maxSel_Date , 101)+'*')
                    where product_id = @product_id and r101_key=@Promo_Key
                    close c_flow
```



```
        deallocate c_flow
        return
    END
END
close c_flow
deallocate c_flow
return
```

## Procedure Name: doavail401keys

```

begin --sp
declare @r401_key varchar(25), @select_date smalldatetime, @bills_mailed numeric(12,2), @product_id
numeric(8,0), @full_promo_key varchar(25), @dt_valid_to smalldatetime, @source_identifier
char(1), @keyCount numeric(8,0), @import_file_type char(1)
declare @fulfillment_house varchar(50)

set nocount on
/*add for product id smarts */
declare c_r401 cursor for /* r401 recordset - the main/reference one */
select r401_key, select_date, bills_mailed, product_id, source_identifier, import_file_type from
r401_summary where not exists
(select * from keys k
where k.full_promo_key = r401_summary.r401_key and k.product_id =
r401_summary.product_id) order by product_id, r401_key
open c_r401
fetch c_r401 into @r401_key, @select_date, @bills_mailed, @product_id, @source_identifier,
@import_file_type

while (@@fetch_status = 0)
begin
Insert into avail_keys (full_promo_key --1
,dt_valid_to --2
,mail_qty --3
,product_id --4
,source_identifier --5
,position1 --6
,position2 --7
,position3 --8
,position4 --9
,position5 --10
,position6 --11
,position7 --12
,position8 --13
,position9 --14
,position10 --15
,fulfillment_house) --16
values
(@r401_key --1
,ISNULL(@select_date, getDate()) --2
,@bills_mailed --3
,@product_id --4
,@source_identifier --5
,SUBSTRING(@r401_key, 1, 1) --6
,SUBSTRING(@r401_key, 2, 1) --7
,SUBSTRING(@r401_key, 3, 1) --8
,SUBSTRING(@r401_key, 4, 1) --9
,SUBSTRING(@r401_key, 5, 1) --10
,SUBSTRING(@r401_key, 6, 1) --11
,SUBSTRING(@r401_key, 7, 1) --12
,SUBSTRING(@r401_key, 8, 1) --13
,SUBSTRING(@r401_key, 9, 1) --14
,SUBSTRING(@r401_key, 10, 1) --15

```



```
, case @import_file_type
    when 'A' then 'CENTROBE'
    when 'B' then 'CENTROBE'
    when 'C' then 'CENTROBE'
    when 'E' then 'CENTROBE'
    when 'F' then 'CDS'
    when 'G' then 'CDS'
    when 'H' then 'CDS'
    when 'I' then 'PALM COAST'
    when 'J' then 'PALM COAST'
    when 'K' then 'PALM COAST'
    else null
end)
```

```
    fetch c_r401 into @r401_key,@select_date,@bills_mailed,@product_id,@source_identifier,
@import_file_type
    end
close c_r401
deallocate c_r401
```

```
delete from avail_keys where exists
    (select * from keys k
    where k.full_promo_key = avail_keys.full_promo_key and k.product_id = avail_keys.product_id)
end --sp
```

```
exec autosetup_wizard 1, -1 --1 to display text (zero suppresses), -1 for all products
```



Procedure Name: dovailkeysbypid

@PID numeric(9,0)

AS

```
declare @r101_key varchar(25), @select_date smalldatetime, @mail_qty numeric(12,2), @product_id
numeric(8,0), @full_promo_key varchar(25), @dt_valid_to smalldatetime, @source_identifier
char(1), @keyCount numeric(8,0)
set nocount on
/*add for product id smarts */
declare c_r101 cursor for /* r101 recordset - the main/reference one */
select distinct r101_key, select_date, mail_qty, product_id, source_identifier from r101 where
product_id = @PID order by product_id
/*declare c_keys cursor for
select count(full_promo_key) from keys where product_id=@product_id and full_promo_key =
@r101_key*/

open c_r101
fetch c_r101 into @r101_key, @select_date, @mail_qty, @product_id, @source_identifier
delete from avail_keys where product_id = @PID /*delete all existing avail_keys from avail_keys list*/

while (@@fetch_status = 0)
begin
    Insert into avail_keys (full_promo_key, dt_valid_to, mail_qty, product_id, source_identifier,
    position1, position2, position3, position4, position5, position6, position7,
    position8, position9, position10 ) values

    (@r101_key, DATEADD(month, 20, ISNULL(@select_date, getDate())), @mail_qty, @product_id, @source_
    identifier,
    SUBSTRING(@r101_key, 1, 1), SUBSTRING(@r101_key, 2, 1), SUBSTRING(@r101_key, 3, 1),
    SUBSTRING(@r101_key, 4, 1), SUBSTRING(@r101_key, 5, 1),
    SUBSTRING(@r101_key, 6, 1), SUBSTRING(@r101_key, 7, 1), SUBSTRING(@r101_key, 8, 1),
    isnull(SUBSTRING(@r101_key, 9, 1), null), ISNULL(SUBSTRING(@r101_key, 10, 1), null) )

    fetch c_r101 into @r101_key, @select_date, @mail_qty, @product_id, @source_identifier
end
close c_r101
deallocate c_r101

delete from avail_keys where exists
(select * from keys k
where k.full_promo_key = avail_keys.full_promo_key and k.product_id = avail_keys.product_id)
print 'Attempting to Execute doAvail401KeysByPID'
execute doavail401keysbypid @PID
return
```

Procedure Name: dovail401keysbypid

@PID numeric(9,0)  
AS

```
declare @r401_key varchar(25), @select_date smalldatetime, @bills_mailed numeric(12,2), @product_id  
numeric(8,0), @full_promo_key varchar(25), @dt_valid_to smalldatetime, @source_identifier  
char(1), @keyCount numeric(8,0)
```

```
set nocount on
```

```
/*add for product id smarts */
```

```
declare c_r401 cursor for /* r401 recordset - the main/reference one */
```

```
select distinct r401_key, select_date, bills_mailed, product_id, source_identifier from r401_summary  
where product_id = @PID order by product_id
```

```
open c_r401
```

```
fetch c_r401 into @r401_key, @select_date, @bills_mailed, @product_id, @source_identifier
```

```
while (@@fetch_status = 0)  
begin
```

```
Insert into avail_keys (full_promo_key, dt_valid_to, mail_qty, product_id, source_identifier,  
position1, position2, position3, position4, position5, position6, position7,  
position8, position9, position10) values
```

```
(@r401_key, DATEADD(month, 20, ISNULL(@select_date, getDate())), @bills_mailed, @product_id, @sour  
ce_identifier,
```

```
SUBSTRING(@r401_key, 1, 1), SUBSTRING(@r401_key, 2, 1), SUBSTRING(@r401_key, 3, 1),  
SUBSTRING(@r401_key, 4, 1), SUBSTRING(@r401_key, 5, 1),  
SUBSTRING(@r401_key, 6, 1), SUBSTRING(@r401_key, 7, 1), SUBSTRING(@r401_key, 8, 1),  
isnull(SUBSTRING(@r401_key, 9, 1), null), ISNULL(SUBSTRING(@r401_key, 10, 1), null) )
```

```
fetch c_r401 into @r401_key, @select_date, @bills_mailed, @product_id, @source_identifier  
end
```

```
close c_r401
```

```
deallocate c_r401
```

```
delete from avail_keys where exists
```

```
(select * from keys k
```

```
where k.full_promo_key = avail_keys.full_promo_key and k.product_id = avail_keys.product_id)
```

```
return
```

## Procedure Name: dovailkeys

```
declare @r101_key varchar(25), @fulfillment_house varchar(50), @select_date smalldatetime, @mail_qty
numeric(12,2), @product_id numeric(8,0), @full_promo_key varchar(25), @dt_valid_to
smalldatetime, @source_identifier char(1), @keyCount numeric(8,0)
set nocount on
/*add for product id smarts */

declare c_r101 cursor for /* r101 recordset - the main/reference one */
select r101_key, select_date, mail_qty, product_id, source_identifier, fulfillment_house from r101 where not
exists
    (select * from keys k
     where k.full_promo_key = r101.r101_key and k.product_id = r101.product_id) order by
product_id, r101_key
/*declare c_keys cursor for
    select count(full_promo_key) from keys where product_id=@product_id and full_promo_key =
    @r101_key*/

open c_r101
fetch c_r101 into @r101_key, @select_date, @mail_qty, @product_id, @source_identifier,
@fulfillment_house
truncate table avail_keys /*delete all existing avail_keys from avail_keys list*/

while (@@fetch_status = 0)
begin
    --print "inserting into avail_keys ==> " + @r101_key
    Insert into avail_keys (full_promo_key, dt_valid_to, mail_qty, product_id, source_identifier,
    position1, position2, position3, position4, position5, position6, position7,
    position8, position9, position10, fulfillment_house) values
    (@r101_key, ISNULL(@select_date, getDate()), @mail_qty, @product_id, @source_identifier,
    SUBSTRING(@r101_key, 1, 1), SUBSTRING(@r101_key, 2, 1), SUBSTRING(@r101_key, 3, 1),
    SUBSTRING(@r101_key, 4, 1), SUBSTRING(@r101_key, 5, 1),
    SUBSTRING(@r101_key, 6, 1), SUBSTRING(@r101_key, 7, 1), SUBSTRING(@r101_key, 8, 1),
    SUBSTRING(@r101_key, 9, 1), SUBSTRING(@r101_key, 10, 1), @fulfillment_house)

    fetch c_r101 into
    @r101_key, @select_date, @mail_qty, @product_id, @source_identifier, @fulfillment_house
end
close c_r101
deallocate c_r101

delete from avail_keys where exists
    (select * from keys k
     where k.full_promo_key = avail_keys.full_promo_key and k.product_id = avail_keys.product_id)

--exec autosetup_wizard
exec doavail401keys
return
```

**Procedure Name: docampesets**

```
declare c_camp cursor for
    select product_id,camp_name, e_set_name from campaigns where e_set_name > ''

declare @product_id decimal(9), @camp_name varchar(250), @e_set_name varchar (50)
open c_camp
fetch c_camp into @product_id, @camp_name, @e_set_name

while (@@fetch_status = 0)
    begin
        /*UPDATE panels SET e_set_name = @e_set_name WHERE CURRENT OF c_panel*/
        update keys set e_set_name=@e_set_name where product_id = @product_id and camp_name =
        @camp_name and (keys.e_set_name < '' or keys.e_set_name is null)
        fetch c_camp into @product_id, @camp_name, @e_set_name
    end

close c_camp
deallocate c_camp

return
```

Procedure Name: docamprsets

```
declare c_campr cursor for
    select product_id,camp_name,r_set_name from campaigns where r_set_name > ''

declare @product_id decimal(9), @camp_name varchar(250), @r_set_name varchar (50)
open c_campr
fetch c_campr into @product_id, @camp_name, @r_set_name

while (@@fetch_status = 0)
    begin
        /*UPDATE panels SET e_set_name = @e_set_name WHERE CURRENT OF c_panel*/
        update keys set r_set_name=@r_set_name where product_id = @product_id and camp_name =
        @camp_name and (keys.r_set_name < '' or keys.r_set_name is null)
        fetch c_campr into @product_id, @camp_name, @r_set_name
    end

close c_campr
deallocate c_campr

return
```

Procedure Name: doestmaxeffort

```
begin
truncate table est_max_EFFORT
INSERT INTO est_max_EFFORT
select est_max1.product_id,
       est_max1.source_identifier,
       est_max1.issue_identifier,
       est_max1.prior_source_identifier,
       min(est_max1.effort_identifier),
       min(est_max1.sum_mail_qty)
FROM v_est_max1 est_max1, v_est_max2 est_max2
WHERE est_max1.sum_mail_qty = est_max2.est_max
and est_max1.source_identifier = est_max2.source_identifier
and est_max1.issue_identifier=est_max2.issue_identifier
and est_max1.prior_source_identifier = est_max2.prior_source_identifier
GROUP by est_max1.product_id,
         est_max1.source_identifier,
         est_max1.issue_identifier,
         est_max1.prior_source_identifier
end

return
```



```
CREATE PROCEDURE doMW AS
declare @product_id numeric(8,0), @stat_name varchar(50)
/* */
declare c_ cursor for
    select product_id from corp_info
declare c_work cursor for
    select r_set_name from revenue_sets where product_id = @product_id and r_set_name =
'Undefined'
open c_
fetch c_ into @product_id

delete from revenue_sets where r_set_name = 'Undefined'
delete from revenue_sets where r_set_name = 'UNDEFINED'

while (@@fetch_status = 0)
    begin
        open c_work
        fetch c_work into @stat_name
        If (@@fetch_status < 0)
            begin
                Insert into revenue_sets (product_id, r_set_name) values (@product_id, 'Undefined')

                close c_work
            end
        else
            begin
                close c_work
            end
        fetch c_ into @product_id
    end
close c_
deallocate c_
deallocate c_work

update campaigns set r_set_name = 'Undefined' where r_set_name is null
update campaigns set r_set_name = 'Undefined' where r_set_name = "
update panels set r_set_name = 'Undefined' where r_set_name is null
update panels set r_set_name = 'Undefined' where r_set_name = "
update keys set r_set_name = 'Undefined' where r_set_name is null
update keys set r_set_name = 'Undefined' where r_set_name = "
return
```



**Procedure Name: doPanelESets**

```
declare c_panel cursor for
    select product_id, panel_name, e_set_name from panels where e_set_name > ''

declare @product_id decimal(9), @panel_name varchar(250), @e_set_name varchar (50)
open c_panel
fetch c_panel into @product_id, @panel_name, @e_set_name

while (@@fetch_status = 0)
begin
    /*UPDATE panels SET e_set_name = @e_set_name WHERE CURRENT OF c_panel*/
    update keys set e_set_name=@e_set_name where product_id = @product_id and panel_name =
    @panel_name and (keys.e_set_name < '' or keys.e_set_name is null)
    fetch c_panel into @product_id, @panel_name, @e_set_name
end

close c_panel
deallocate c_panel

return
```

**Procedure Name: doPanelRsets**

```
declare c_panelr cursor for
    select product_id, panel_name, r_set_name from panels where r_set_name > ''

declare @product_id decimal(9), @panel_name varchar(250), @r_set_name varchar (50)
open c_panelr
fetch c_panelr into @product_id, @panel_name, @r_set_name

while (@@fetch_status = 0)
begin
    /*UPDATE panels SET e_set_name = @e_set_name WHERE CURRENT OF c_panel*/
    update keys set r_set_name=@r_set_name where product_id = @product_id and panel_name =
    @panel_name and (keys.r_set_name < '' or keys.r_set_name is null)
    fetch c_panelr into @product_id, @panel_name, @r_set_name
end

close c_panelr
deallocate c_panelr

return
```



Procedure Name: doProdSetup

```

/***** Object: Stored Procedure dbo.doProdSetup  Script Date: 1/14/99 3:43:07 PM *****/
/***** Object: Stored Procedure dbo.doProdSetup  Script Date: 1/12/99 11:05:37 PM *****/
/***** Object: Stored Procedure dbo.doProdSetup  Script Date: 11/28/98 2:31:13 PM *****/
/***** Object: Stored Procedure dbo.doProdSetup  Script Date: 11/17/98 2:43:36 PM *****/

declare @product_id numeric(8,0), @grp_name varchar(50), @title_name varchar(50), @src_prod_id
numeric(8,0)
/* */
declare c_corp_info cursor for
    select product_id,grp_name,title_name from corp_info
declare c_src_category cursor for
    select product_id from src_category where product_id = @product_id
open c_corp_info
fetch c_corp_info into @product_id,@grp_name,@title_name
/*if (@@fetch_status <> 0)
    begin
    end*/
while (@@fetch_status = 0)
    begin
        open c_src_category
        fetch c_src_category into @src_prod_id
        If (@@fetch_status <> 0)
            begin
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'Agents'," ,@grp_name,@title_name)
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'Renewals','R things',@grp_name,@title_name)
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'Billing'," ,@grp_name,@title_name)
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'DTP'," ,@grp_name,@title_name)
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'Gifts'," ,@grp_name,@title_name)
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'Renewals'," ,@grp_name,@title_name)
                Insert into src_category (product_id,src_cat_name,src_cat_descr,grp_name,title_name)
values (@product_id,'UNNASSIGNED'," ,@grp_name,@title_name)
                Insert into source
(product_id,src_cat_name,source_name,src_indicator,grp_name,title_name) values
(@product_id,'Renewals','RenewalsSRC','R',@grp_name,@title_name)
            end
        close c_src_category
    end
    else
        begin
            close c_src_category
        end
    fetch c_corp_info into @product_id,@grp_name,@title_name
end
close c_corp_info
deallocate c_corp_info
deallocate c_src_category
return
```





Procedure Name: doR101\_summary

```
begin
truncate table r101

--update r101_flow set select_date = getDate() where select_date < '1901-01-01'

/*
update r101_flow set select_date = getDate() where not exists
(select * from r101_flow r
where r.r101_key = r101_flow.r101_key and r.product_id = r101_flow.product_id and select_date
> '1990-01-01')
*/

Insert into r101
(r101_reorg_date,
product_id,
r101_key,
source_identifier,
issue_identifier,
prior_source_identifier,
effort_identifier,
mail_qty,
select_date,
key_descr,
subs_this_week,
subs_to_date,
gross_subs,
gross_subs_pct,
net_subs,
net_subs_pct,
tot_net_subs,
tot_net_subs_pct,
pct_credit,
pct_cred_renew,
cred_pay_amt,
cred_pay_pct,
avg_net_trm,
avg_net_trm_value,
avg_tot_net_trm,
avg_tot_net_trm_value,
subs_phone,
subs_c_card,
install_1st,
install_1st_total,
subs_gift,
subs_gift_pct_1st_trm,
sub_years_gross,
sub_years_gross_tot_net,
gross_sub_revenue,
net_sub_revenue,
gross_sub_copies,
net_sub_copies,
cash_subs,
```

```
est_max_eff_mail_qty,  
gross_cred_subs,  
net_cred_subs,  
fulfillment_title_name,  
gross_list_rental_qty,  
net_list_rental_qty,  
package_identifier,  
key_descr2,  
donees_todate,  
total_orders,  
tot_order_pct_rspnce,  
renew_subs_pct  
, fulfillment_house)
```

```
EXECUTE("select r101_reorg_date,  
product_id,  
r101_key,  
source_identifier,  
issue_identifier,  
prior_source_identifier,  
effort_identifier,  
mail_qty,  
select_date,  
key_descr,  
subs_this_week,  
subs_to_date,  
gross_subs,  
gross_subs_pct,  
net_subs,  
net_subs_pct,  
tot_net_subs,  
tot_net_subs_pct,  
pct_credit,  
pct_cred_renew,  
cred_pay_amt,  
cred_pay_pct,  
avg_net_trm,  
avg_net_trm_value,  
avg_tot_net_trm,  
avg_tot_net_trm_value,  
subs_phone,  
subs_c_card,  
install_1st,  
install_1st_total,  
subs_gift,  
subs_gift_pct_1st_trm,  
sub_years_gross,  
sub_years_gross_tot_net,  
gross_sub_revenue,  
net_sub_revenue,  
gross_sub_copies,  
net_sub_copies,  
cash_subs,  
est_max_eff_mail_qty,  
gross_cred_subs,
```



```
net_cred_subs,
fulfillment_title_name,
gross_list_rental_qty,
net_list_rental_qty,
package_identifier,
key_descr2,
donees_todate,
total_orders,
tot_order_pct_rspnce,
renew_subs_pct
, isnull(case import_file_type
          when 'A' then 'CENTROBE'
          when 'B' then 'CENTROBE'
          when 'C' then 'CENTROBE'
          when 'E' then 'CENTROBE'
          when 'F' then 'CDS'
          when 'G' then 'CDS'
          when 'H' then 'CDS'
          when 'I' then 'PALM COAST'
          when 'J' then 'PALM COAST'
          when 'K' then 'PALM COAST'
          else null
        end
        ,fulfillment_house)
from vr101_flow_max_dates, r101_flow
where mproduct_id = product_id and
mr101_key=r101_key and mdate=r101_reorg_date
order by mproduct_id, mr101_key, r101_reorg_date")
end
```



Procedure Name: doR401\_summary

truncate table r401\_summary

update r401\_flow set select\_date = getDate() where not exists  
(select \* from r401\_flow r  
where r.r401\_key = r401\_flow.r401\_key and r.product\_id = r401\_flow.product\_id and select\_date  
> '1990-01-01')

Insert into r401\_summary  
(r401\_reorg\_date,  
product\_id,  
r401\_key,  
source\_identifer,  
credit\_period\_identifer,  
current\_source\_identifer,  
effort\_identifer,  
select\_date,  
bills\_mailed,  
bills\_mailed\_amt,  
this\_week\_pmt,  
total\_pmt,  
pmt\_pct,  
step\_up,  
step\_up\_pct,  
pmt\_plus\_step\_up\_rev,  
avg\_rev\_per\_pmt,  
avg\_term\_per\_pmt,  
request\_cancel,  
request\_cancel\_pct,  
per\_cancel,  
per\_cancel\_pct,  
pmt\_on\_pmt,  
pmt\_after\_cancel,  
total\_copies,  
fulfillment\_title\_name,  
est\_max\_eff\_mail\_qty  
, import\_file\_type)  
EXECUTE(  
SELECT distinct r401\_reorg\_date,  
product\_id,  
r401\_key,  
source\_identifer,  
credit\_period\_identifer,  
current\_source\_identifer,  
effort\_identifer,  
select\_date,  
bills\_mailed,  
bills\_mailed\_amt,  
this\_week\_pmt,  
total\_pmt,  
pmt\_pct,  
step\_up,



```
step_up_pct,  
pmt_plus_step_up_rev,  
avg_rev_per_pmt,  
avg_term_per_pmt,  
request_cancel,  
request_cancel_pct,  
per_cancel,  
per_cancel_pct,  
pmt_on_pmt,  
pmt_after_cancel,  
total_copies,  
fulfillment_title_name,  
est_max_eff_mail_qty  
, import_file_type  
from vr401_flow_max_dates, r401_flow where mproduct_id = product_id and  
mr401_key=r401_key and mdate=r401_reorg_date order by product_id, r401_key, r401_reorg_date  
)
```



## Procedure Name: doRecycleKey

```
@product_id_r numeric(8,2),
@promo_key_r varchar (25)
AS
```

```
declare @r401_key varchar(25),
        @select_date smalldatetime,
        @mail_qty numeric(12,2),
        @product_id numeric(8,0),
        @full_promo_key varchar(25),
        @dt_valid_to smalldatetime,
        @source_identifier char(1),
        @keyCount numeric(8,0),
        @src_cat_name VARCHAR(50),
        @r401_key varchar(25),
        @bills_mailed numeric(9),
        @fulfillment_house varchar(50),
        @import_file_type varchar(50)
```

```
select @src_cat_name = src_cat_name
from keys where full_promo_key = @promo_key_r
and product_id = @product_id_r
```

```
if @src_cat_name = 'Billing'
```

```
BEGIN
```

```
print @src_cat_name
```

```
declare c_r401 cursor for /* r401 recordset - the main/reference one */
```

```
select r401_key,select_date,bills_mailed,product_id,source_identifier, import_file_type from
r401_summary
```

```
where r401_key = @promo_key_r
```

```
and product_id = @product_id_r
```

```
open c_r401
```

```
fetch c_r401 into @r401_key,@select_date,@bills_mailed,@product_id,@source_identifier,
```

```
@import_file_type
```

```
while (@@fetch_status = 0)
```

```
begin
```

```
Insert into avail_keys (full_promo_key,dt_valid_to,mail_qty,product_id,source_identifier,
```

```
position1,position2,position3,position4,position5,position6, position7,
```

```
position8,position9,position10, fulfillment_house ) values
```

```
(@r401_key,ISNULL(@select_date,getDate()),@bills_mailed,@product_id,@source_identifier,
```

```
SUBSTRING(@r401_key,1,1),SUBSTRING(@r401_key,2,1),SUBSTRING(@r401_key,3,1),
```

```
SUBSTRING(@r401_key,4,1),SUBSTRING(@r401_key,5,1),
```

```
SUBSTRING(@r401_key,6,1),SUBSTRING(@r401_key,7,1),SUBSTRING(@r401_key,8,1),
```

```
isnull(SUBSTRING(@r401_key,9,1), null),
```

```
ISNULL(SUBSTRING(@r401_key,10,1),null)
```

```
, case @import_file_type
```

```
when 'A' then 'CENTROBE'
```

```
when 'B' then 'CENTROBE'
```

```
when 'C' then 'CENTROBE'
```

```
when 'E' then 'CENTROBE'
```

```
when 'F' then 'CDS'
```

```
when 'G' then 'CDS'
```



```
        when 'H' then 'CDS'
        when 'I' then 'PALM COAST'
        when 'J' then 'PALM COAST'
        when 'K' then 'PALM COAST'
    end)
    fetch c_r401 into @r401_key,@select_date,@bills_mailed,@product_id,@source_identifier,
    @import_file_type
    end
close c_r401
deallocate c_r401
--delete from avail_keys where exists
--(select * from keys k
--where k.full_promo_key = avail_keys.full_promo_key and k.product_id =
avail_keys.product_id)
return
END

ELSE

BEGIN
print '101stuff'
set nocount on
/*add for product id smarts */
declare c_r101_r cursor for /* r101 recordset - the main/reference one */
select r101_key,select_date,mail_qty,product_id,source_identifier, fulfillment_house from r101 where
product_id = @product_id_r and r101_key = @promo_key_r
open c_r101_r
fetch c_r101_r into @r101_key,@select_date,@mail_qty,@product_id,@source_identifier,
@import_file_type
while (@@fetch_status = 0)
    begin
        Insert into avail_keys
        (full_promo_key,dt_valid_to,mail_qty,product_id,source_identifier,position1,position2,position3,position4,
        position5,position6, position7, fulfillment_house)
        values
        (@r101_key,ISNULL(@select_date,getDate()),@mail_qty,@product_id,@source_identifier,SUBSTRING
        (@r101_key,1,1),SUBSTRING(@r101_key,2,1),SUBSTRING(@r101_key,3,1),SUBSTRING(@r101_key
        ,4,1),SUBSTRING(@r101_key,5,1),SUBSTRING(@r101_key,6
        ,1),SUBSTRING(@r101_key,7,1)
        , @import_file_type)
        fetch c_r101_r into @r101_key,@select_date,@mail_qty,@product_id,@source_identifier,
        @import_file_type
        end
    close c_r101_r
    deallocate c_r101_r
    return
END
```



## Procedure Name: doRecyclePanelKey

```
@rec_id numeric(8,2)
AS
/* get primary key info */
declare @camp_name    varchar(250),
        @source_name  varchar(50),
        @panel_name   varchar(250),
        @src_cat_name varchar(30),
        @product_id   numeric(5),
        @keys_product_id numeric(5),
        @full_promo_key numeric(5)

SELECT      @camp_name = camp_name,
            @source_name = source_name,
            @panel_name = panel_name,
            @src_cat_name = src_cat_name,
            @product_id = product_id
FROM panels
WHERE rec_id = @rec_id

declare c_keys cursor for
select full_promo_key, product_id
from keys
where product_id = @product_id
and camp_name = @camp_name
and source_name = @source_name
and panel_name = @panel_name
and src_cat_name = @src_cat_name

open c_keys
fetch c_keys into @full_promo_key, @keys_product_id

while (@@fetch_status = 0)
begin
    exec doRecycleKey @keys_product_id, @full_promo_key
    fetch c_keys into @full_promo_key, @keys_product_id
end
close c_keys
deallocate c_keys
delete from keys
where product_id = @product_id
and camp_name = @camp_name
and source_name = @source_name
and panel_name = @panel_name
and src_cat_name = @src_cat_name

return
```



Procedure Name: doSourceInds

```
/** creates a src_indicator placeholder from 101 flow **/  
declare @var_src_ind char(1), @var_target_ind char(1)  
declare c_r101_ind cursor for  
    select distinct source_identifier from r101  
declare c_source cursor for  
    select src_indicator from source where src_indicator = @var_src_ind  
open c_r101_ind  
fetch c_r101_ind into @var_src_ind  
if (@@fetch_status <> 0)  
    begin  
        print "No Source Indicators found in r101"  
        close c_r101_ind  
        deallocate c_r101_ind  
        return  
    end  
while (@@fetch_status = 0)  
    begin  
        open c_source  
        fetch c_source into @var_target_ind  
        If (@@fetch_status <> 0)  
            begin  
                Insert into source (src_cat_name,src_indicator,source_name) values  
                ('UNASSIGNED',@var_src_ind,'MISSING')  
            close c_source  
            end  
        else  
            begin  
                close c_source  
            end  
        fetch c_r101_ind into @var_src_ind  
    end  
close c_r101_ind  
deallocate c_r101_ind  
return
```



Procedure Name: doSummary

```
BEGIN -- only for Palm Coast titles at Primedia
update r401_flow set select_date = r401_reorg_date where select_date = '1900-01-01'
END
```

```
BEGIN
PRINT 'Executing dor101_summary'
end
```

```
BEGIN
execute dor101_summary;
PRINT 'Execution of dor101_summary complete'
END
```

```
BEGIN
PRINT 'Executing dor401_summary'
END
```

```
BEGIN
execute dor401_summary;
PRINT 'Execution of dor401_summary complete'
END
```

```
BEGIN
PRINT 'Executing doavailkeys'
END
```

```
BEGIN
execute doavailkeys;
PRINT 'Execution of doavailkeys complete'
END
```

```
BEGIN
PRINT 'Executing doavail401keys'
END
```

--\*\*\*\*\* doavail401keys runs when doavailkeys runs !

```
--BEGIN
--execute doavail401keys;
--PRINT 'Execution of doavail401keys complete'
--END
```

```
BEGIN
execute doEstMaxEffort ;
PRINT 'Execution of doavail401keys complete'
END
```

```
Begin
Insert into summary_log values(GetDate(), 'Summary Procedure Run')
Print 'Updated summary log'
```



END



Procedure Name: getCorpInfo

```
/****** Object: Stored Procedure dbo.getCorpInfo  Script Date: 1/14/99 3:43:06 PM *****/  
/****** Object: Stored Procedure dbo.getCorpInfo  Script Date: 1/12/99 11:05:37 PM *****/  
(  
  @product_id numeric(8,0)  
)  
AS  
select * from corp_info where product_id = @product_id  
return
```





## Procedure Name: expenses

```
/****** Object: Stored Procedure dbo.expenses Script Date: 1/14/99 3:43:07 PM *****/
/****** Object: Stored Procedure dbo.expenses Script Date: 1/12/99 11:05:38 PM *****/
/****** Object: Stored Procedure dbo.expenses Script Date: 11/28/98 2:31:13 PM *****/
/****** Object: Stored Procedure dbo.expenses Script Date: 11/17/98 2:43:36 PM *****/
/****** Object: Stored Procedure dbo.expenses Script Date: 10/19/98 6:03:58 PM *****/
```

```
declare @e_set_name varchar (50),@product_id numeric(8,2),
        @print_cost numeric(11, 3),@print_cpm_cpu varchar (10),
        @print_basis varchar (20),
        @ltr_cost numeric(11, 3),@ltr_cpm_cpu varchar (10),
        @ltr_basis varchar (20),
        @postage_out_cost numeric(11, 3),@postage_out_cpm_cpu varchar (10),
        @postage_out_basis varchar (20),
        @postage_in_cost numeric(11, 3),@postage_in_cpm_cpu varchar (10),
        @postage_in_basis varchar (20),
        @premium_cost numeric(11, 3),@premium_cpm_cpu varchar (10),
        @premium_basis varchar (20),
        @badpay_cost numeric(11, 3),@badpay_cpm_cpu varchar (10),
        @badpay_basis varchar (20),
        @billing_cost numeric(11, 3),@billing_cpm_cpu varchar (10),
        @billing_basis varchar (20),
        @subs_svc_costs numeric(11, 3),@subs_svc_cpm_cpu varchar (10),
        @subs_svc_basis varchar (20),
        @exp_fixed_costs numeric(11, 3),@exp_fixed_cpm_cpu varchar (10),
        @exp_fixed_basis varchar (20),
        @exp_other_costs numeric(11, 3),@exp_other_cpm_cpu varchar (10),
        @exp_other_basis varchar (20),

        @full_promo_key varchar (10),
        @level1_expense numeric(12,3),
        @level2_expense numeric(12,3),
        @level3_expense numeric(12,3),
        @r101_key varchar(10),
        @mail_qty numeric(12,2),
        @gross_subs numeric(7,2),
        @net_subs integer,

        @l1_tmp numeric (12,3),
        @l1_value numeric (12,3),
        @l2_tmp numeric (12,3),
        @l2_value numeric (12,3),
        @l3_tmp numeric (12,3),
        @l3_value numeric (12,3)

declare c_exp cursor for
        select expense_sets.e_set_name, expense_sets.product_id,
        print_cost,print_cpm_cpu,print_basis,
        ltr_cost,ltr_cpm_cpu,ltr_basis,
        /*deleted list_cost,cpm and basis because of redesign-list library activities.. moved to key, still
        need to add code for calculating */
        postage_out_cost,postage_out_cpm_cpu,postage_out_basis,
        postage_in_cost,postage_in_cpm_cpu,postage_in_basis,
        premium_cost,premium_cpm_cpu,premium_basis,
```

```

badpay_cost,badpay_cpm_cpu,badpay_basis,
billing_cost,billing_cpm_cpu,billing_basis,
subs_svc_costs,subs_svc_cpm_cpu,subs_svc_basis,
exp_fixed_costs,exp_fixed_cpm_cpu,exp_fixed_basis,
exp_other_costs,exp_other_cpm_cpu,exp_other_basis,
full_promo_key,
level1_revenue,
level2_revenue,
level3_revenue,
r101_key,
mail_qty,
gross_subs,
net_subs
from expense_sets,keys,r101 where keys.product_id=expense_sets.product_id and
keys.product_id=r101.product_id and
keys.e_set_name = expense_sets.e_set_name and keys.full_promo_key = r101.r101_key
Order by expense_sets.e_set_name
open c_exp
fetch c_exp into
    @e_set_name,@product_id,
    @print_cost,@print_cpm_cpu,
    @print_basis,
    @ltr_cost,@ltr_cpm_cpu,
    @ltr_basis,
    @postage_out_cost,@postage_out_cpm_cpu,
    @postage_out_basis,
    @postage_in_cost,@postage_in_cpm_cpu,
    @postage_in_basis,
    @premium_cost,@premium_cpm_cpu,
    @premium_basis,
    @badpay_cost,@badpay_cpm_cpu,
    @badpay_basis,
    @billing_cost,@billing_cpm_cpu,
    @billing_basis,
    @subs_svc_costs,@subs_svc_cpm_cpu,
    @subs_svc_basis,
    @exp_fixed_costs,@exp_fixed_cpm_cpu,
    @exp_fixed_basis,
    @exp_other_costs,@exp_other_cpm_cpu,
    @exp_other_basis,

    @full_promo_key,
    @level1_expense,
    @level2_expense,
    @level3_expense,
    @r101_key,
    @mail_qty,

    @gross_subs,
    @net_subs
while (@@fetch_status = 0)
begin
    select @l1_tmp = 0
    select @l1_value = 0
    select @l2_tmp = 0

```

```
select @l2_value = 0
select @l3_tmp = 0
select @l3_value = 0
/*****LEVEL 1 *****/
If (@print_basis = 'Mail Qty')
begin
select @l1_tmp = (@mail_qty * @print_cost)
end
else if (@print_basis = 'Gross Subs')
begin
select @l1_tmp = (@gross_subs * @print_cost)
end
else if (@print_basis = 'Net Subs')
begin
select @l1_tmp = (@net_subs * @print_cost)
end
if (@print_cpm_cpu = 'CPM')
begin
select @l1_tmp = @l1_tmp / 1000
end
select @l1_value = @l1_value + @l1_tmp

select @l1_tmp = 0
If (@ltr_basis = 'Mail Qty')
begin
select @l1_tmp = (@mail_qty * @ltr_cost)
end
else if (@ltr_basis = 'Gross Subs')
begin
select @l1_tmp = (@gross_subs * @ltr_cost)
end
else if (@ltr_basis = 'Net Subs')
begin
select @l1_tmp = (@net_subs * @ltr_cost)
end
if (@ltr_cpm_cpu = 'CPM')
begin
select @l1_tmp = @l1_tmp / 1000
end
select @l1_value = @l1_value + @l1_tmp

select @l1_tmp = 0
If (@postage_out_basis = 'Mail Qty')
begin
select @l1_tmp = (@mail_qty * @postage_out_cost)
end
else if (@postage_out_basis = 'Gross Subs')
begin
select @l1_tmp = (@gross_subs * @postage_out_cost)
end
else if (@postage_out_basis = 'Net Subs')
begin
select @l1_tmp = (@net_subs * @postage_out_cost)
end
if (@postage_out_cpm_cpu = 'CPM')
```



```
begin

    select @l1_tmp = @l1_tmp / 1000
    end
select @l1_value= @l1_value + @l1_tmp
/*****LEVEL 1*****/
/*****LEVEL 2*****/

If (@postage_in_basis = 'Mail Qty')
begin
    select @l2_tmp = (@mail_qty * @postage_in_cost)
    end
else if (@postage_in_basis = 'Gross Subs')
begin
    select @l2_tmp = (@gross_subs * @postage_in_cost)
    end
else if (@postage_in_basis = 'Net Subs')
begin
    select @l2_tmp = (@net_subs * @postage_in_cost)
    end
if (@postage_in_cpm_cpu = 'CPM')
begin
    select @l2_tmp = @l2_tmp / 1000
    end
select @l2_value= @l2_value + @l2_tmp

select @l2_tmp = 0
If (@premium_basis = 'Mail Qty')
begin
    select @l2_tmp = (@mail_qty * @premium_cost)
    end
else if (@premium_basis = 'Gross Subs')
begin
    select @l2_tmp = (@gross_subs * @premium_cost)
    end
else if (@premium_basis = 'Net Subs')
begin
    select @l2_tmp = (@net_subs * @premium_cost)
    end
if (@premium_cpm_cpu = 'CPM')
begin
    select @l2_tmp = @l2_tmp / 1000
    end
select @l2_value= @l2_value + @l2_tmp
select @l2_tmp = 0
If (@badpay_basis = 'Gross - Net Subs')
begin
    select @l2_tmp = ((@gross_subs - @net_subs) * @badpay_cost)
    end
if (@badpay_cpm_cpu = 'CPM')
begin
    select @l2_tmp = @l2_tmp / 1000
    end
select @l2_value= @l2_value + @l2_tmp
```



```
select @l2_tmp = 0
If (@billing_basis = 'Mail Qty')
begin
select @l2_tmp = (@mail_qty * @billing_cost)
end
else if (@billing_basis = 'Gross Subs')
begin
select @l2_tmp = (@gross_subs * @billing_cost)
end
else if (@billing_basis = 'Net Subs')
begin
select @l2_tmp = (@net_subs * @billing_cost)
end
if (@billing_cpm_cpu = 'CPM')
begin
select @l2_tmp = @l2_tmp / 1000
end
select @l2_value= @l2_value + @l2_tmp
/*****LEVEL 2*****/
/*****LEVEL 3*****/
If (@subs_svc_basis = 'Mail Qty')
begin
select @l3_tmp = (@mail_qty * @subs_svc_costs)
end
else if (@subs_svc_basis = 'Gross Subs')
begin
select @l3_tmp = (@gross_subs * @subs_svc_costs)
end
else if (@subs_svc_basis = 'Net Subs')
begin
select @l3_tmp = (@net_subs * @subs_svc_costs)
end
if (@subs_svc_cpm_cpu = 'CPM')
begin
select @l3_tmp = @l3_tmp / 1000
end
select @l3_value= @l3_value + @l3_tmp

select @l3_tmp = 0
If (@exp_fixed_basis = 'Mail Qty')
begin
select @l3_tmp = (@mail_qty * @exp_fixed_costs)
end
else if (@exp_fixed_basis = 'Gross Subs')
begin
select @l3_tmp = (@gross_subs * @exp_fixed_costs)
end
else if (@exp_fixed_basis = 'Net Subs')
begin

select @l3_tmp = (@net_subs * @exp_fixed_costs)
end
if (@exp_fixed_cpm_cpu = 'CPM')
begin
select @l3_tmp = @l3_tmp / 1000
```



```
        end
select @l3_value= @l3_value + @l3_tmp
        select @l3_tmp = 0
        If (@exp_other_basis = 'Mail Qty')
            begin
                select @l3_tmp = (@mail_qty * @exp_other_costs)
            end
        else if (@exp_other_basis = 'Gross Subs')
            begin
                select @l3_tmp = (@gross_subs * @exp_other_costs)
            end
        else if (@exp_other_basis = 'Net Subs')
            begin
                select @l3_tmp = (@net_subs * @exp_other_costs)
            end
        if (@exp_other_cpm_cpu = 'CPM')
            begin
                select @l3_tmp = @l3_tmp / 1000
            end
select @l3_value= @l3_value + @l3_tmp
/*****LEVEL 3*****/
        UPDATE keys SET level1_expense = @l1_value,level2_expense = @l2_value,level3_expense =
        @l3_value WHERE CURRENT OF c_exp
fetch c_exp into
        @e_set_name,@product_id,
        @print_cost,@print_cpm_cpu,
        @print_basis,
        @ltr_cost,@ltr_cpm_cpu,
        @ltr_basis,
        @postage_out_cost,@postage_out_cpm_cpu,
        @postage_out_basis,
        @postage_in_cost,@postage_in_cpm_cpu,
        @postage_in_basis,
        @premium_cost,@premium_cpm_cpu,
        @premium_basis,
        @badpay_cost,@badpay_cpm_cpu,
        @badpay_basis,
        @billing_cost,@billing_cpm_cpu,
        @billing_basis,
        @subs_svc_costs,@subs_svc_cpm_cpu,
        @subs_svc_basis,
        @exp_fixed_costs,@exp_fixed_cpm_cpu,
        @exp_fixed_basis,
        @exp_other_costs,@exp_other_cpm_cpu,
        @exp_other_basis,

        @full_promo_key,
        @level1_expense,
        @level2_expense,
        @level3_expense,
        @r101_key,
        @mail_qty,
        @gross_subs,
        @net_subs
end
```



```
close c_exp  
deallocate c_exp  
return
```



**Procedure Name: maxBillEffort**

```
update r401_flow set est_max_eff_mail_qty = bills_mailed where effort_identifier = '1'  
update r401_summary set est_max_eff_mail_qty =bills_mailed where effort_identifier = '1'  
update r401_flow set est_max_eff_mail_qty = 0 where est_max_eff_mail_qty is null  
update r401_summary set est_max_eff_mail_qty = 0 where est_max_eff_mail_qty is null
```



**Procedure Name: maxEffort**

```
update r101_flow set est_max_eff_mail_qty = mail_qty where effort_identifier = 1
update r101 set est_max_eff_mail_qty = mail_qty where effort_identifier = 1
update r101_flow set est_max_eff_mail_qty = 0 where est_max_eff_mail_qty is null
update r101 set est_max_eff_mail_qty = 0 where est_max_eff_mail_qty is null
```



Procedure Name: maxGiftEffort

```
update r301_flow set est_max_eff_mail_qty = mail_qty where effort_identifier = '1'  
update r301_summary set est_max_eff_mail_qty = mail_qty where effort_identifier = '1'  
update r301_flow set est_max_eff_mail_qty = 0 where est_max_eff_mail_qty is null  
update r301_summary set est_max_eff_mail_qty = 0 where est_max_eff_mail_qty is null
```



Procedure Name: procedure Rebuild\_Rpt\_Cats

begin

```
-- UPDATES IMAGINE REPORTS
truncate table imagine.dbo.rpt_cats
insert into imagine.dbo.rpt_cats
    select src_cat_name, src_sub_cat, location, rpt_name, rpt_description, rpt_filters from
pmedia.dbo.rpt_cats
BEGIN
    PRINT 'IMAGINE Report Catalog Updated'
END

-- UPDATES IMT PRESENTATION REPORTS
truncate table IMTPresentation.dbo.rpt_cats
insert into IMTPresentation.dbo.rpt_cats
    select src_cat_name, src_sub_cat, location, rpt_name, rpt_description, rpt_filters from
pmedia.dbo.rpt_cats
BEGIN
    PRINT 'IMT Presentation Report Catalog Updated'
END

-- UPDATES DWELL REPORTS
truncate table dwell.dbo.rpt_cats
insert into dwell.dbo.rpt_cats
    select src_cat_name, src_sub_cat, location, rpt_name, rpt_description, rpt_filters from
pmedia.dbo.rpt_cats
BEGIN
    PRINT 'Dwell Report Catalog Updated'
END

-- UPDATES DWELL REPORTS
truncate table demo.dbo.rpt_cats
insert into demo.dbo.rpt_cats
    select src_cat_name, src_sub_cat, location, rpt_name, rpt_description, rpt_filters from
pmedia.dbo.rpt_cats
BEGIN
    PRINT 'Dwell Report Catalog Updated'
END
```

End



Procedure Name: sp\_add\_autosetup\_scheme

```
@product_id int
, @src_cat_name varchar(50)
, @src_indicator varchar(3)
, @source_name varchar (250)
, @valid_from datetime
, @valid_to datetime
, @setup_mask varchar(50)
, @e_set_name varchar(50)

AS
declare @title varchar(50)
declare @grp_name varchar(50)

select @title = title_name
      , @grp_name = grp_name
      from corp_info
      where product_id = @product_id
--gw added next line
if @e_set_name is null set @e_set_name = 'Undefined'

insert into autosetup_scheme
(
    product_id
    , title
    , grp_name
    , src_cat_name
    , src_indicator
    , source_name
    , valid_from_date
    , valid_to_date
    , setup_mask
    , e_set_name
)
values
(
    @product_id
    , @title
    , @grp_name
    , @src_cat_name
    , @src_indicator
    , @source_name
    , @valid_from
    , @valid_to
    , @setup_mask
    , @e_set_name
)
```



Procedure Name: sp\_Add\_Campaign

```
@camp_name varchar(250), -- 1
@camp_type varchar(50), -- 2
@camp_status_name varchar(20), -- 3
@camp_date_fr smalldatetime, -- 4
@camp_date_to smalldatetime, -- 5
@agt_code varchar(10), -- 6
@agt_descr varchar(50), -- 7
@e_set_name varchar(50), -- 8
@r_set_name varchar(50), -- 9
@newstand_sale numeric(18), -- 10
@budgeted_gross_pct numeric(9), -- 11
@budgeted_vol numeric(9), -- 12
```

```
@source_name varchar (250), -- 13
@src_indicator char (3), -- 14
@product_id numeric(8,0), -- 15
@src_cat_name varchar(30), -- 16
@grp_name varchar(50), -- 17
@title_name varchar (250) -- 18
```

AS

insert into campaigns

```
(
    camp_name,
    camp_type,
    camp_status_name,
    camp_date_fr,
    camp_date_to,
    agt_code,
    agt_descr,
    e_set_name,
    r_set_name,
    newstand_sale,
    budgeted_vol,
    budgeted_gross_pct,

    product_id,
    src_cat_name,
    source_name,
    src_indicator,
    grp_name,
    title_name
)
values
(
    @camp_name,
    isNull( @camp_type,'Undefined'),
    isNull( @camp_status_name,'Undefined'),
    @camp_date_fr,
```



```
@camp_date_to,  
@agt_code,  
@agt_descr,  
isnull(@e_set_name,'Undefined'),  
isnull(@r_set_name,'Undefined'),  
@newstand_sale,  
@budgeted_vol,  
@budgeted_gross_pct,  
  
@product_id,  
@src_cat_name,  
@source_name,  
@src_indicator,  
@grp_name,  
@title_name  
)
```



Procedure Name: sp\_add\_campaign\_def

```
@product_id int
, @campaign_code varchar(50)
, @campaign_description varchar(50)
, @src_cat_name varchar(50)
, @campaign_date datetime
, @panel_scheme varchar(50)
```

AS

insert into autosetup\_campaign\_def

```
(
    product_id
    , campaign_code
    , campaign_description
    , src_cat_name
    , campaign_date
    , panel_scheme
)
values
(
    @product_id
    , @campaign_code
    , @campaign_description
    , @src_cat_name
    , @campaign_date
    , @panel_scheme
)
```



Procedure Name: sp\_Add\_Corp\_Panel\_Sub\_Type

```
@panel_sub_type_name varchar(50),  
@panel_sub_type_descr varchar(250)
```

AS

```
insert into Corp_Panel_Sub_Type
```

```
(
```

```
    panel_sub_type_name,  
    panel_sub_type_descr
```

```
)
```

```
values
```

```
(
```

```
    @panel_sub_type_name,  
    @panel_sub_type_descr
```

```
)
```





Procedure Name: sp\_Add\_Corp\_Panel\_Test\_Type

```
@panel_test_type_name varchar(50),  
@panel_test_type_descr varchar(250)
```

AS

```
insert into Corp_Panel_Test_Type
```

```
(  
    panel_test_type_name,  
    panel_test_type_descr
```

```
)
```

```
values
```

```
(  
    @panel_test_type_name,  
    @panel_test_type_descr  
)
```



Procedure Name: sp\_Add\_Corp\_Panel\_Type

```
@panel_type_name varchar(50),  
@panel_type_descr varchar(250)
```

AS

insert into Corp\_Panel\_Type

(

```
    panel_type_name,  
    panel_type_descr
```

)

values

(

```
    @panel_type_name,  
    @panel_type_descr
```

)



Procedure Name: sp\_Add\_Keys

```
@rec_id numeric(9,0),
@src_cat_name varchar(50),
@source_name varchar(250),
@src_indicator char(3),
@camp_name varchar(250),
@panel_name varchar(250)
```

AS

```
declare @product_id numeric(8,0),      --for keys table
        @full_promo_key varchar(25),
        @dt_valid_from smalldatetime,
        @dt_valid_to smalldatetime,
        @key_mail_qty numeric(8,0),
        @list_cost_basis varchar(25),
        @list_cat_name varchar(250),
        @list_name varchar(250),
        @list_segment_name varchar(250),
                                -- for info from panels table
        @PANEL_TYPE varchar(50),
        @PANEL_SUBTYPE varchar(50),
        @TEST_TYPE varchar(50),
        @PANEL_OFFER varchar(50),
        @PANEL_OFFER_PRICE money,
        @PANEL_PKG varchar(50),
        @PANEL_PREMIUM varchar(50),
        @PANEL_DATE_START smalldatetime,
        @PANEL_DATE_END smalldatetime,
        @GRP_NAME varchar(50),
        @TITLE_NAME varchar(50),
        @CAMP_TYPE varchar(50),
        @CAMP_STATUS_NAME varchar(50),
        @AGT_CODE varchar(50),
        @AGT_DESCR varchar(50),
        @CAMP_DATE_FR smalldatetime,
        @CAMP_DATE_TO smalldatetime,
        @E_SET_NAME varchar(50),
        @R_SET_NAME varchar(50),
        @newstand_sale_marker varchar(50),
        @newstand_sale NUMERIC(18)

select @full_promo_key = full_promo_key,
       @dt_valid_to = dt_valid_to,
       @product_id = product_id,
       @key_mail_qty = mail_qty
from avail_keys
where rec_id=@rec_id

--get panel and other info for key table
declare c_panel_info cursor for
select PANEL_TYPE ,
```



```
PANEL_SUBTYPE ,
TEST_TYPE ,
PANEL_OFFER,
PANEL_OFFER_PRICE ,
PANEL_PKG ,
PANEL_PREMIUM ,
PANEL_DATE_START,
PANEL_DATE_END ,
GRP_NAME ,
TITLE_NAME ,
CAMP_TYPE ,
CAMP_STATUS_NAME ,
AGT_CODE ,
AGT_DESCR ,
CAMP_DATE_FR ,
CAMP_DATE_TO, E_SET_NAME, R_SET_NAME
from panels where product_id = @product_id and src_cat_name = @src_cat_name and
source_name = @source_name
and camp_name = @camp_name and panel_name = @panel_name
select @PANEL_TYPE = PANEL_TYPE ,
@PANEL_SUBTYPE = PANEL_SUBTYPE ,
@TEST_TYPE = TEST_TYPE ,
@PANEL_OFFER = PANEL_OFFER,
@PANEL_OFFER_PRICE = PANEL_OFFER_PRICE ,
@PANEL_PKG = PANEL_PKG ,
@PANEL_PREMIUM = PANEL_PREMIUM ,
@PANEL_DATE_START = PANEL_DATE_START,
@PANEL_DATE_END = PANEL_DATE_END,
@GRP_NAME = GRP_NAME,
@TITLE_NAME = TITLE_NAME ,
@CAMP_TYPE = CAMP_TYPE ,
@CAMP_STATUS_NAME = CAMP_STATUS_NAME ,
@AGT_CODE = AGT_CODE ,
@AGT_DESCR = AGT_DESCR ,
@CAMP_DATE_FR = CAMP_DATE_FR ,
@CAMP_DATE_TO = CAMP_DATE_TO,
@E_SET_NAME = E_SET_NAME,
@R_SET_NAME = R_SET_NAME ,
@newstand_sale_marker = newstand_sale_marker,
@newstand_sale = Newstand_sale
from panels
where product_id = @product_id
and src_cat_name = @src_cat_name
and source_name = @source_name
and camp_name = @camp_name
and panel_name = @panel_name

insert into keys
(
    full_promo_key,

    dt_valid_from,
    dt_valid_to,
    src_cat_name,
    src_indicator,
```

panel\_name,  
camp\_name,  
product\_id,  
source\_name,  
key\_mail\_qty,

PANEL\_TYPE,  
PANEL\_SUBTYPE,  
TEST\_TYPE,  
PANEL\_OFFER,  
PANEL\_OFFER\_PRICE,  
PANEL\_PKG,  
PANEL\_PREMIUM,  
PANEL\_DATE\_START,  
PANEL\_DATE\_END,  
GRP\_NAME,  
TITLE\_NAME,  
CAMP\_TYPE,  
CAMP\_STATUS\_NAME,  
AGT\_CODE,  
AGT\_DESCR,  
CAMP\_DATE\_FR,  
CAMP\_DATE\_TO,  
E\_SET\_NAME,  
R\_SET\_NAME,  
newstand\_sale\_marker,  
newstand\_sale,  
list\_cost\_basis,  
list\_cat\_name,  
list\_name,  
list\_segment\_name

)

values

( @full\_promo\_key,  
dateadd(month,-20,@dt\_valid\_to),  
@dt\_valid\_to,  
@src\_cat\_name,  
@src\_indicator,  
@panel\_name,  
@camp\_name,  
@product\_id,  
@source\_name,  
@key\_mail\_qty,  
  
@PANEL\_TYPE,  
@PANEL\_SUBTYPE,  
@TEST\_TYPE,  
@PANEL\_OFFER,  
@PANEL\_OFFER\_PRICE,  
isNull(@PANEL\_PKG,'Undefined'),  
@PANEL\_PREMIUM,  
@PANEL\_DATE\_START,  
@PANEL\_DATE\_END,  
@GRP\_NAME,  
@TITLE\_NAME,

```
@CAMP_TYPE,  
@CAMP_STATUS_NAME,  
@AGT_CODE,  
@AGT_DESCR,  
@CAMP_DATE_FR,  
@CAMP_DATE_TO,  
'Undefined',  
'Undefined',  
isNull(@newstand_sale_marker,'N'),  
@newstand_sale,  
'Mail QTY',  
'Undefined',  
'Undefined',  
'Undefined'  
)
```

```
if (@@error = 0)  
begin  
Delete from avail_keys where rec_id = @rec_id  
end  
close c_panel_info  
deallocate c_panel_info
```

Procedure Name: sp\_Add\_Keys2

```
@rec_id numeric(9,0),
@src_cat_name varchar(30),
@source_name varchar(250),
@src_indicator char(3),
@camp_name varchar(250),
@panel_name varchar(250)
```

AS

```
declare @product_id numeric(8,0),      --for keys table
        @full_promo_key varchar(25),
        @dt_valid_from smalldatetime,
        @dt_valid_to smalldatetime,
        @key_mail_qty numeric(8,0),
        -- for info from panels table
        @PANEL_TYPE varchar(50),
        @PANEL_SUBTYPE varchar(50),
        @TEST_TYPE varchar(50),
        @PANEL_OFFER varchar(50),
        @PANEL_OFFER_PRICE money,
        @PANEL_PKG varchar(50),
        @PANEL_PREMIUM varchar(50),
        @PANEL_DATE_START smalldatetime,
        @PANEL_DATE_END smalldatetime,
        @GRP_NAME varchar(50),
        @TITLE_NAME varchar(50),
        @CAMP_TYPE varchar(50),
        @CAMP_STATUS_NAME varchar(50),
        @AGT_CODE varchar(50),
        @AGT_DESCR varchar(50),
        @CAMP_DATE_FR smalldatetime,
        @CAMP_DATE_TO smalldatetime
```

```
declare c_avail key cursor for
        select full_promo_key, dt_valid_to, product_id, mail_qty from avail_keys where rec_id=@rec_id
```

```
open c_avail_key
fetch c_avail_key into @full_promo_key, @dt_valid_to, @product_id, @key_mail_qty
```

```
--get panel and other info for key table
declare c_panel_info cursor for
        select PANEL_TYPE ,
        PANEL_SUBTYPE ,
        TEST_TYPE ,
        PANEL_OFFER,
        PANEL_OFFER_PRICE ,
        PANEL_PKG ,
        PANEL_PREMIUM ,
        PANEL_DATE_START,
```



```
PANEL_DATE_END ,
GRP_NAME ,
TITLE_NAME ,
CAMP_TYPE ,
CAMP_STATUS_NAME ,
AGT_CODE ,
AGT_DESCR ,
CAMP_DATE_FR ,
CAMP_DATE_TO from panels where product_id = @product_id and src_cat_name =
@src_cat_name and source_name = @source_name
and camp_name = @camp_name and panel_name = @panel_name
```

```
open c_panel_info
```

```
fetch c_panel_info into
```

```
@PANEL_TYPE,
@PANEL_SUBTYPE,
@TEST_TYPE,
@PANEL_OFFER,
@PANEL_OFFER_PRICE,
@PANEL_PKG,
@PANEL_PREMIUM,
@PANEL_DATE_START,
@PANEL_DATE_END,
@GRP_NAME,
@TITLE_NAME,
@CAMP_TYPE,
@CAMP_STATUS_NAME,
@AGT_CODE,
@AGT_DESCR,
@CAMP_DATE_FR,
@CAMP_DATE_TO
```

```
print dateadd(month,-20,@dt_valid_to)
```

```
insert into keys
```

```
(
```

```
full_promo_key,
```

```
dt_valid_from,
```

```
dt_valid_to,
```

```
src_cat_name,
```

```
src_indicator,
```

```
panel_name,
```

```
camp_name,
```

```
product_id,
```

```
source_name,
```

```
key_mail_qty,
```

```
PANEL_TYPE,
```

```
PANEL_SUBTYPE,
```

```
TEST_TYPE,
```

```
PANEL_OFFER,
```

```
PANEL_OFFER_PRICE,
```

```
PANEL_PKG,
```

```
PANEL_PREMIUM,
```

```
PANEL_DATE_START,
```





```
PANEL_DATE_END,
GRP_NAME,
TITLE_NAME,
CAMP_TYPE,
CAMP_STATUS_NAME,
AGT_CODE,
AGT_DESCR,
CAMP_DATE_FR,
CAMP_DATE_TO

)
values
(
    @full_promo_key,
    dateadd(month,-20,@dt_valid_to),
    @dt_valid_to,
    @src_cat_name,
    @src_indicator,
    @panel_name,
    @camp_name,
    @product_id,
    @source_name,
    @key_mail_qty,

    @PANEL_TYPE,
    @PANEL_SUBTYPE,
    @TEST_TYPE,
    @PANEL_OFFER,
    @PANEL_OFFER_PRICE,
    @PANEL_PKG,
    @PANEL_PREMIUM,
    isNull( @PANEL_DATE_START,null) ,
    isNull(@PANEL_DATE_END,null) ,
    @GRP_NAME,
    @TITLE_NAME,
    @CAMP_TYPE,
    @CAMP_STATUS_NAME,
    @AGT_CODE,
    @AGT_DESCR,
    isNull(@CAMP_DATE_FR,null) ,
    isNull( @CAMP_DATE_TO,null)

)

close c_avail_key
deallocate c_avail_key

Delete from avail_keys where rec_id = @rec_id

close c_panel_info
deallocate c_panel_info
```



Procedure Name: sp\_Add\_Panel

```
@panel_name varchar(250), --1
@panel_type varchar(50), --2
@panel_subtype varchar(50), --3
@test_type varchar(50), --4
@panel_pkg varchar(50), --5
@panel_date_start datetime, --6
@panel_date_end datetime, --7
@e_set_name varchar(50), --8
@r_set_name varchar(50), --9
@newstand_sale_marker char(1), --10
```

```
@product_id numeric(8,0), --11
@grp_name varchar(50), --12
@title_name varchar(250), --13
@src_cat_name varchar(30), --14
@source_name varchar(250), --15
@src_indicator char(3), --16
@camp_name varchar(250), --17
@camp_type varchar(50), --18
@camp_status_name varchar(20), --19
@agt_code varchar(10), --20
@agt_descr varchar(50), --21
@camp_date_fr datetime, --22
@camp_date_to datetime --23
```

AS  
declare

```
@IE_SET_NAME varchar(50),
@IR_SET_NAME varchar(50),
@lNewStand_sale NUMERIC(18)
```

--get expense and revenue set names for pass down if blank

```
select @IE_SET_NAME = E_SET_NAME,
@IR_SET_NAME = R_SET_NAME,
@lNewStand_sale = Newstand_sale
from campaigns where product_id = @PRODUCT_ID
and SRC_CAT_NAME = @SRC_CAT_NAME and SOURCE_NAME = @SOURCE_NAME
and CAMP_NAME = @CAMP_NAME
```

insert into panels

```
(
    panel_name,
    panel_type,
    panel_subtype,
    test_type,
    panel_pkg,
    panel_date_start,
    panel_date_end,
    e_set_name,
```

```
r_set_name,
newstand_sale_marker,

product_id,
grp_name,
title_name,
src_cat_name,
source_name,
src_indicator,
camp_name,
camp_type,
camp_status_name,
agt_code,
agt_descr,
camp_date_fr,
camp_date_to,
Newstand_sale
)
values
(

@panel_name,
isnull(@panel_type,'Undefined'),
isnull(@panel_subtype,'Undefined'),
isnull(@test_type,'Undefined'),
isnull(@panel_pkg,'Undefined'),
@panel_date_start,
@panel_date_end,
isnull(@e_set_name,@IE_SET_NAME),
isnull(@r_set_name,@IR_SET_NAME),
@newstand_sale_marker,

@product_id,
@grp_name,
@title_name,
@src_cat_name,
@source_name,
@src_indicator,
@camp_name,
@camp_type,
@camp_status_name,
@agt_code,
@agt_descr,
@camp_date_fr,
@camp_date_to,
@lNewStand_sale
)
```



Procedure Name: sp\_add\_panel\_def

```
@product_id int
, @panel_scheme varchar(50)
, @panel_codes varchar(250)
, @panel_name varchar(50)
```

AS

insert into autosetup\_panel\_def

```
(
    product_id
    , panel_scheme
    , panel_codes
    , panel_name
)
```

values

```
(
    @product_id
    , @panel_scheme
    , @panel_codes
    , @panel_name
)
```

---

Procedure Name: sp\_Add\_prior\_Source\_desc

```
    @prior_source_identifier char (2)
    , @prior_source_desc varchar (20)
    , @billing_cur_src_desc varchar (255)
    , @model_source_cat varchar(50)
    , @model_source varchar(50)
    , @agent_src_desc varchar(50)
    , @valid_from_date datetime
    , @valid_to_date datetime
    , @product_id numeric(18, 0)
```

as

insert into Prior\_source\_desc

```
(
    product_id
    , prior_source_identifier
    , prior_source_desc
    , billing_cur_src_desc
    , model_source_cat
    , model_source
    , agent_src_desc
    , valid_from_date
    , valid_to_date
)
values (
    @product_id
    , @prior_source_identifier
    , @prior_source_desc
    , @billing_cur_src_desc
    , @model_source_cat
    , @model_source
    , @agent_src_desc
    , @valid_from_date
    , @valid_to_date
)
```

Procedure Name: sp\_Add\_Prod\_Camp\_Status

@camp\_status\_name varchar(20),  
@camp\_status\_descr varchar(250),

@product\_id numeric(8,0)

AS

insert into Prod\_Camp\_Status

(  
    camp\_status\_name,  
    camp\_status\_descr,  
    product\_id

)  
values

(  
    @camp\_status\_name,  
    @camp\_status\_descr,  
    @product\_id  
)



Procedure Name: sp\_Add\_Prod\_Camp\_Type

@camp\_type\_name varchar(250),  
@camp\_type\_descr varchar(250),

@product\_id numeric(8,0)

AS

insert into Prod\_Camp\_Type

(

camp\_type\_name,  
camp\_type\_descr,  
product\_id

)

values

(

@camp\_type\_name,  
@camp\_type\_descr,  
@product\_id

)

Procedure Name: sp\_Add\_Prod\_E\_Set

```
@product_id numeric(8, 0) ,  
@e_set_name varchar (50) ,  
@print_cost numeric(11, 3) ,  
@print_cpm_cpu varchar (10) ,  
@print_basis varchar (20) ,  
@merge_purge numeric(11, 3) ,  
@merge_cpm_cpu varchar (10) ,  
@merge_basis varchar (20) ,  
@ltr_cost numeric(11, 3) ,  
@ltr_cpm_cpu varchar (10) ,  
@ltr_basis varchar (20) ,  
@postage_out_cost numeric(11, 3) ,  
@postage_out_cpm_cpu varchar (10) ,  
@postage_out_basis varchar (20) ,  
@postage_in_cost numeric(11, 3) ,  
@postage_in_cpm_cpu varchar (10) ,  
@postage_in_basis varchar (20) ,  
@premium_cost numeric(11, 3) ,  
@premium_cpm_cpu varchar (10) ,  
@premium_basis varchar (20) ,  
@badpay_cost numeric(11, 3) ,  
@badpay_cpm_cpu varchar (10) ,  
@badpay_basis varchar (20) ,  
@billing_cost numeric(11, 3) ,  
@billing_cpm_cpu varchar (10) ,  
@billing_basis varchar (20) ,  
@subs_svc_costs numeric(11, 3) ,  
@subs_svc_cpm_cpu varchar (10) ,  
@subs_svc_basis varchar (20) ,  
@exp_other_costs numeric(11, 3) ,  
@exp_other_cpm_cpu varchar (10) ,  
@exp_other_basis varchar (20)
```

AS

insert into expense\_sets

(

```
product_id ,  
e_set_name ,  
print_cost ,  
print_cpm_cpu ,  
print_basis ,  
merge_purge ,  
merge_cpm_cpu ,  
merge_basis ,  
ltr_cost ,  
ltr_cpm_cpu ,  
ltr_basis ,  
postage_out_cost ,  
postage_out_cpm_cpu ,  
postage_out_basis ,  
postage_in_cost ,  
postage_in_cpm_cpu ,
```



```
postage_in_basis ,
premium_cost,
premium_cpm_cpu,
premium_basis ,
badpay_cost ,
badpay_cpm_cpu ,
badpay_basis ,
billing_cost ,
billing_cpm_cpu ,
billing_basis ,
subs_svc_costs ,
subs_svc_cpm_cpu ,
subs_svc_basis ,
exp_other_costs ,
exp_other_cpm_cpu ,
exp_other_basis
)
values
(
    @product_id ,
    @e_set_name ,
    @print_cost ,
    @print_cpm_cpu ,
    @print_basis,
    @merge_purge,
    @merge_cpm_cpu,
    @merge_basis,
    @ltr_cost ,
    @ltr_cpm_cpu ,
    @ltr_basis ,
    @postage_out_cost ,
    @postage_out_cpm_cpu ,
    @postage_out_basis ,
    @postage_in_cost ,
    @postage_in_cpm_cpu ,
    @postage_in_basis ,
    @premium_cost,
    @premium_cpm_cpu,
    @premium_basis ,
    @badpay_cost ,
    @badpay_cpm_cpu ,
    @badpay_basis ,
    @billing_cost ,
    @billing_cpm_cpu ,
    @billing_basis ,
    @subs_svc_costs ,
    @subs_svc_cpm_cpu ,
    @subs_svc_basis ,
    @exp_other_costs ,
    @exp_other_cpm_cpu ,
    @exp_other_basis
)
```

Procedure Name: sp\_Add\_Prod\_List\_cat

-- added on 01/05/2000

@name varchar(250),  
@descr varchar(250),  
@product\_id numeric(8,0)

insert into List\_cats

```
(  
    list_cat_name,  
    list_cat_descr,  
    product_id  
)  
values  
(  
    @name,  
    @descr,  
    @product_id  
)
```



Procedure Name: sp\_Add\_Prod\_List\_Name

-- added on 01/05/2000

```
@name varchar(250),  
@descr varchar(250),  
@product_id numeric(8,0)
```

AS

insert into List\_names

```
(  
    list_name,  
    list_name_description,  
    product_id
```

```
)
```

values

```
(  
    @name,  
    @descr,  
    @product_id  
)
```



Procedure Name: sp\_Add\_Prod\_List\_Segment

-- added on 01/05/2000

```
@name varchar(250),  
@descr varchar(250),  
@product_id numeric(8,0)
```

AS

insert into List\_Segments

```
(  
    list_segment_name,  
    list_segment_description,  
    product_id  
)  
values  
(  
    @name,  
    @descr,  
    @product_id  
)
```



Procedure Name: sp\_Add\_Prod\_Panel\_Package

```
@package_name varchar(50),  
@package_description varchar(250),  
@Product_id numeric(9)
```

AS

insert into Prod\_panel\_Package

```
( product_id,  
    package_name,  
    package_description
```

```
)
```

values

```
( @product_id,  
    @package_name,  
    @package_description
```

```
)
```



Procedure Name: sp\_Add\_Prod\_R\_Set

```
@product_id numeric(8, 0),
@r_set_name varchar (50) ,
@subs_revenue numeric(11, 3) ,
@subs_cpm_cpu varchar (10) ,
@ad_revenue numeric(11, 3),
@ad_cpm_cpu varchar (10),
@listrental_revenue numeric(11, 3) ,
@listrental_cpm_cpu varchar (10),
@other_revenue numeric(11, 3),
@other_cpm_cpu varchar (10)
```

AS

insert into revenue\_sets

```
( product_id ,
    r_set_name ,
    subs_revenue ,
    subs_cpm_cpu ,
    ad_revenue ,
    ad_cpm_cpu ,
    listrental_revenue ,
    listrental_cpm_cpu ,
    other_revenue ,
    other_cpm_cpu )
```

values

```
( @product_id ,
    @r_set_name ,
    @subs_revenue ,
    @subs_cpm_cpu ,
    @ad_revenue ,
    @ad_cpm_cpu ,
    @listrental_revenue ,
    @listrental_cpm_cpu ,
    @other_revenue ,
    @other_cpm_cpu )
```



Procedure Name: sp\_add\_renewal\_psdef

```
    @product_id int
    , @src_cat_name varchar(50)
    , @src_indicator varchar(50)
    , @prior_src_indicator varchar(250)
    , @source_name varchar (250)
    , @valid_from datetime
    , @valid_to datetime
```

AS

```
declare @title varchar(50)
```

```
declare @grp_name varchar(50)
```

```
select @title = title_name
      from corp_info
      where product_id = @product_id
```

```
insert into autosetup_renewal_psdef
```

```
(
    product_id
    , title
    , src_cat_name
    , source_indicator
    , prior_source_indicator
    , source_name
    , valid_from_date
    , valid_to_date
)
```

values

```
(
    @product_id
    , @title
    , @src_cat_name
    , @src_indicator
    , @prior_src_indicator
    , @source_name
    , @valid_from
    , @valid_to
)
```



Procedure Name: sp\_Add\_Source

@source\_name varchar (250),  
@src\_indicator char (3),

@product\_id numeric(8,0),  
@src\_cat\_name varchar(30),  
@grp\_name varchar(50),  
@title\_name varchar (250)

AS

insert into source

```
(
    product_id,
    src_cat_name,
    source_name,
    src_indicator,
    grp_name,
    title_name
)
values
(
    @product_id,
    @src_cat_name,
    @source_name,
    @src_indicator,
    @grp_name,
    @title_name
)
```



Procedure Name: sp\_Add\_Source\_description

```
        @u_source_name varchar(250),
        @u_source_description varchar(250)
AS
insert into Source_descriptions
(
    u_source_name,
    u_source_description
)
values
(
    @u_source_name,
    @u_source_description
)
```

Procedure Name: sp\_cascade\_product\_id

```
INSERT INTO list_cats
SELECT product_id, 'Undefined',"
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM list_cats
  WHERE list_cat_name = 'Undefined' )
```

```
INSERT INTO list_names
SELECT product_id, 'Undefined',"
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM list_names
  WHERE list_name = 'Undefined' )
```

```
INSERT INTO list_segments
SELECT product_id, 'Undefined',"
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM list_segments
  WHERE list_segment_name = 'Undefined' )
```

```
INSERT INTO expense_sets
SELECT product_id, 'Undefined',
  null, null, null, null,
  null, null, null, null,
  null, null, null, null,
  null, null, null, null,
  null, null, null, null,
  null, null, null, null,
  null, null, null, null,
  null
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM expense_sets
  WHERE e_set_name = 'Undefined' )
```

```
INSERT INTO revenue_sets
SELECT product_id, 'Undefined',
  null, null, null, null,
  null, null, null, null
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM revenue_sets
  WHERE r_set_name = 'Undefined' )
```

```
INSERT INTO prod_camp_status
SELECT product_id, 'Undefined',"
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM prod_camp_status
  WHERE camp_status_name = 'Undefined' )
```

```
INSERT INTO prod_camp_type
SELECT product_id, 'Undefined',"
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM prod_camp_type
  WHERE camp_type_name = 'Undefined' )
```

```
INSERT INTO prod_panel_package
SELECT product_id, 'Undefined',"
FROM corp_info
WHERE product_id not in (
  select product_id
  FROM prod_panel_package
  WHERE package_name = 'Undefined' )
```



Procedure Name: sp\_DeleteCamp\_Statues

@product\_id numeric(9),  
@camp\_status\_name varchar(20)

AS

delete from camp\_statuses where product\_id = @product\_id and camp\_status\_name =  
@camp\_status\_name



Procedure Name: sp\_DeleteCamp\_Types

@product\_id numeric(9),  
@camp\_type varchar(50)

AS

delete from camp\_types where product\_id = @product\_id and camp\_type = @camp\_type



Procedure Name: sp\_DeletePanel\_SubTypes

@product\_id numeric(9),  
@panel\_subtype varchar(50)

AS

delete from p\_sub\_types where product\_id = @product\_id and panel\_subtype = @panel\_subtype



Procedure Name: sp\_DeletePanel\_TestTypes

@product\_id numeric(9),  
@test\_type varchar(50)

AS

delete from p\_test\_types where product\_id = @product\_id and test\_type = @test\_type



Procedure Name: sp\_DeletePanel\_Types

@product\_id numeric(9),  
@panel\_type varchar(50)

AS

delete from p\_types where product\_id = @product\_id and panel\_type = @panel\_type





Procedure Name: sp\_DeleteRPT\_Def

@rpt\_id numeric(9)  
AS

delete from rpt\_Defs where rpt\_id = @rpt\_id



Procedure Name: sp\_edit\_autosetup

```
@src_cat_name varchar(50)
, @src_indicator varchar(10)
, @source_name varchar(250)
, @valid_from_date datetime
, @valid_to_date datetime
, @setup_mask varchar(50)
, @e_set_name varchar(50)
, @login Varchar(50)
, @rec_id numeric(8,0)
, @isDelete numeric(1,0)
```

AS

```
If (@isDelete=1)
begin
    delete from autosetup_scheme where rec_id = @rec_id
end
if (@isDelete=0)
begin
    UPDATE autosetup_scheme
    SET src_cat_name = @src_cat_name
      , source_name = @source_name
      , src_indicator = @src_indicator
      , valid_from_date = @valid_from_date
      , valid_to_date = @valid_to_date
      , setup_mask = @setup_mask
      , e_set_name = @e_set_name
    WHERE rec_id = @rec_id
end
```



Procedure Name: sp\_edit\_autosetup\_campaign\_def

```
@campaign_code varchar(50)
, @campaign_description varchar(200)
, @src_cat_name varchar(250)
, @campaign_date datetime
, @panel_scheme varchar(50)
, @rec_id numeric(8,0)
, @isDelete numeric(1,0)
```

AS

```
If (@isDelete=1)
begin
    delete from autosetup_campaign_def where rec_id = @rec_id
end
if (@isDelete=0)
begin
    UPDATE autosetup_campaign_def
    SET campaign_code = @campaign_code
      , campaign_description = @campaign_description
      , src_cat_name = @src_cat_name
      , campaign_date = @campaign_date
      , panel_scheme = @panel_scheme
    WHERE rec_id = @rec_id
End
```



Procedure Name: sp\_edit\_autosetup\_panel\_def

```
@panel_scheme varchar(50)
, @panel_codes varchar(255)
, @panel_name varchar(50)
, @rec_id numeric(8,0)
, @isDelete numeric(1,0)
```

AS

```
If (@isDelete=1)
begin
    delete from autosetup_panel_def where rec_id = @rec_id
end
if (@isDelete=0)
begin
    UPDATE autosetup_panel_def
    SET panel_scheme = @panel_scheme
        , panel_codes = @panel_codes
        , panel_name = @panel_name
    WHERE rec_id = @rec_id
end
```

Procedure Name: procedure sp\_edit\_autosetup\_renewal\_psdef

```
@src_cat_name varchar(50)
, @source_name varchar(50)
, @source_indicator varchar(50)
, @prior_source_indicator varchar(255)
, @valid_from_date datetime
, @valid_to_date datetime
, @rec_id numeric(8,0)
, @isDelete numeric(1,0)
```

AS

```
If (@isDelete=1)
begin
    delete from autosetup_renewal_psdef where rec_id = @rec_id
end
if (@isDelete=0)
begin
    UPDATE autosetup_renewal_psdef
    SET src_cat_name = @src_cat_name
      , source_name = @source_name
      , source_indicator = @source_indicator
      , prior_source_indicator = @prior_source_indicator
      , valid_from_date = @valid_from_date
      , valid_to_date = @valid_to_date
    WHERE rec_id = @rec_id
end
```

Procedure Name: sp\_Edit\_Campaign

```

@camp_name varchar(250), -- 1
@camp_date_fr smalldatetime, -- 2
@camp_type varchar(50), -- 3
@camp_status_name varchar(20), -- 4
--@camp_date_to smalldatetime, -- 5
--@agt_code varchar(10), -- 6
--@agt_descr varchar(50), -- 7
@e_set_name varchar(50), -- 8
@r_set_name varchar(50), -- 9
--@newstand_sale numeric(18), -- 10
@budgeted_gross_pct numeric(18,3), -- 11
@budgeted_vol numeric(18,3), -- 12
@login VARCHAR(50), -- 13

@rec_id numeric(8,0),
@isDelete numeric(1,0)

```

AS

declare

```

@IPRODUCT_ID numeric(8,0),
@ICAMP_NAME varchar(50),
@ISOURCE_NAME varchar(250),
@IE_set_name Varchar(50),
@IR_set_name Varchar(50),
@lsrc_cat_name VARCHAR(30),
@message VARCHAR(255)

```

--Copy vars

```

declare @copycampname varchar(50)
declare @copies int

```

--get camp, panel and other info for deletions from key table

begin --sp

```

select  @IPRODUCT_ID = product_id,
        @ICAMP_NAME = camp_name,
        @ISOURCE_NAME = source_name,
        @lsrc_cat_name = src_cat_name ,
        @IE_set_name = E_set_name,
        @IR_set_name = r_set_name

```

from campaigns

where rec\_id = @rec\_id

if (@IsDelete=2) --then copy existing campaign and associated records into new record(s)

begin

```

set @copycampname = "Copy __ of " + @camp_name

```

--Get number of next copy

```

select @copies = (count(*) + 1)

```

from campaigns

where camp\_name like @copycampname

```

if @copies < 10 set @copycampname = "Copy " + cast(@copies as varchar(1)) + " of "

```

```

else
    set @copycampname = "Copy " + cast(@copies as varchar(2)) + " of "

```

```
--Insert new copy of campaign
insert campaigns (src_indicator,
                  src_cat_name,
                  product_id,
                  camp_type,
                  camp_name,
                  camp_status_name,
                  --agt_code,
                  --agt_descr,
                  camp_date_fr,
                  --camp_date_to,
                  e_set_name,
                  r_set_name,
                  grp_name,
                  title_name,
                  source_name,
                  newstand_sale,
                  budgeted_gross_pct,
                  budgeted_vol,
                  campaign_code)

--values
select            src_indicator,
                  src_cat_name,
                  product_id,
                  camp_type,
                  @copycampname + @camp_name as campname,
                  camp_status_name,
                  --agt_code,
                  --agt_descr,
                  camp_date_fr,
                  --dateadd(month,1,camp_date_to) as camp_date_to,
                  e_set_name,
                  r_set_name,
                  grp_name,
                  title_name,
                  source_name,
                  newstand_sale,
                  budgeted_gross_pct,
                  budgeted_vol,
                  campaign_code
                  from campaigns
                  where rec_id = @rec_id

--Insert new copies of panels from the campaign we are copying
insert panels
select @copycampname + @camp_name as camp_name,
       src_indicator,
       panel_name,
       src_cat_name,
       product_id,
       panel_type,
       e_set_name,
       panel_subtype,
       r_set_name,
       test_type,
```

```
        panel_offer,
        panel_offer_price,
        panel_pkg,
        panel_premium,
        panel_date_start,
        panel_date_end,
        grp_name,
        title_name,
        source_name,
        camp_type,
        camp_status_name,
        agt_code,
        agt_descr,
        camp_date_fr,
        camp_date_to,
        Newstand_Sale_Marker,
        Newstand_Sale,
        Budgeted_gross_pct,
        Budgeted_vol,
        campaign_code
        , panel_code
        , null as order_qty
        , null as dist_qty
    from panels
    where product_id = @lPRODUCT_ID
    and src_cat_name = @lsrc_cat_name
    and source_name = @lSOURCE_NAME
    and camp_name = @lcAMP_NAME

end --if
If (@isDelete=1) --then delete campaign and associated records
begin

    declare c_keys cursor for /* r101 recordset - the main/reference one */
    select product_id, full_promo_key
    from keys
    where product_id = @lproduct_id
    and camp_name = @lcamp_name
    and SOURCE_NAME = @lSOURCE_NAME
    and src_cat_name = @lsrc_cat_name

    declare
        @lkeys_PRODUCT_ID numeric(8,0),
        @lfull_promo_key varchar(50)

    open c_keys
    fetch c_keys into @lkeys_product_id, @lfull_promo_key

    while (@@fetch_status = 0)
    begin
        exec doRecycleKey @lkeys_product_id, @lfull_promo_key
        fetch c_keys into @lkeys_product_id, @lfull_promo_key
    end

    close c_keys
    deallocate c_keys
```



```
delete
from keys
where product_id = @PRODUCT_ID
and src_cat_name = @src_cat_name
and source_name = @SOURCE_NAME
and camp_name = @CAMP_NAME

delete
from panels
where product_id = @PRODUCT_ID
and src_cat_name = @src_cat_name
and source_name = @SOURCE_NAME
and camp_name = @CAMP_NAME

delete from campaigns where rec_id = @rec_id
set @message = 'Campaign: ' + @CAMP_NAME + ' Source: ' + @SOURCE_NAME + ' Src_cat: ' +
@src_cat_name + ', delete applied to panel and keys'
exec sp_logininfo @product_id, @login, 'Delete Campaign', @message
end
if (@IsDelete=0) --Update
begin
    update keys
    set camp_name = @camp_name,
        camp_type=isNull(@camp_type,'Undefined'),
        camp_status_name = isNull( @camp_status_name,'Undefined'),
        camp_date_fr=@camp_date_fr,
        --camp_date_to=@camp_date_to,
        --agt_code=@agt_code,
        --agt_descr=@agt_descr,
        --Newstand_sale = @newStand_sale,
        budgeted_vol = @budgeted_vol, -- 11
        budgeted_gross_pct = @budgeted_gross_pct
    where product_id = @PRODUCT_ID
    and source_name = @SOURCE_NAME
    and camp_name = @CAMP_NAME
    and src_cat_name = @src_cat_name

    update panels
    set camp_name = @camp_name,
        camp_type=isNull(@camp_type,'Undefined'),
        camp_status_name = isNull( @camp_status_name,'Undefined'),
        camp_date_fr=@camp_date_fr,
        --camp_date_to=@camp_date_to,
        --agt_code=@agt_code,
        --agt_descr=@agt_descr,
        --Newstand_sale = @newStand_sale,
        budgeted_vol = @budgeted_vol, -- 11
        budgeted_gross_pct = @budgeted_gross_pct
    where product_id = @PRODUCT_ID
    and source_name = @SOURCE_NAME
    and camp_name = @CAMP_NAME
    and src_cat_name = @src_cat_name
```

```

update campaigns
set camp_name = @camp_name,
  camp_type=isNull(@camp_type,'Undefined'),
  camp_status_name = isNull( @camp_status_name,'Undefined'),
  camp_date_fr=@camp_date_fr,
  --camp_date_to=@camp_date_to,
  --agt_code=@agt_code,
  --agt_descr=@agt_descr,
  e_set_name=isNull(@e_set_name,'Undefined'),
  r_set_name=isNull(@r_set_name,'Undefined'),
  --newstand_sale = @newstand_sale,
  budgeted_vol = @budgeted_vol,    -- 11
  budgeted_gross_pct = @budgeted_gross_pct
  where rec_id = @rec_id
if (@e_Set_name <> @lE_set_name )
BEGIN
  update panels
  set e_set_name = @e_Set_name
  WHERE product_id = @lPRODUCT_ID
  and source_name = @lSOURCE_NAME
  and camp_name = @lCAMP_NAME
  and e_set_name = @le_Set_name
  and src_cat_name = @lsrc_cat_name

  update keys
  set e_set_name = @e_Set_name
  WHERE product_id = @lPRODUCT_ID
  and source_name = @lSOURCE_NAME
  and camp_name = @lCAMP_NAME
  and e_set_name = @le_Set_name
  and src_cat_name = @lsrc_cat_name
END
if (@r_Set_name <> @lr_set_name )
BEGIN
  update panels
  set r_set_name = @r_Set_name
  WHERE product_id = @lPRODUCT_ID
  and source_name = @lSOURCE_NAME
  and camp_name = @lCAMP_NAME
  and r_set_name = @lr_Set_name
  and src_cat_name = @lsrc_cat_name
  update keys
  set r_set_name = @r_Set_name
  WHERE product_id = @lPRODUCT_ID
  and source_name = @lSOURCE_NAME
  and camp_name = @lCAMP_NAME
  and r_set_name = @lr_Set_name
  and src_cat_name = @lsrc_cat_name
END
close c_camp_info2
deallocate c_camp_info2
if @lcamp_name <> @camp_name
BEGIN
  set @message = 'Old Campaign: ' + @lCAMP_NAME + ', New Campaign: ' +
    @CAMP_NAME + ', Source: ' +

```



```
        @ISOURCE_NAME + ', Src_cat: ' + @lsrc_cat_name + ', edit applied to panel and keys'
    END
    ELSE
    BEGIN
        set @message = 'Campaign: ' + @lCAMP_NAME + ', Source: ' +
        @ISOURCE_NAME + ', Src_cat: ' + @lsrc_cat_name + ', edit applied to panel and keys'
    END
    exec sp_loginfo @lproduct_id, @login, 'Edit Campaign', @message
end --if
end --sp
```



Procedure Name: sp\_Edit\_Corp\_Panel\_Sub\_Type

```
@panel_sub_type_name varchar(50),
@panel_sub_type_descr varchar(250),

@rec_id numeric(8,0),
@isDelete numeric(1,0)
AS

declare @old_value varchar(50)
select @old_value = panel_sub_type_name
  From corp_panel_sub_type
 where rec_id = @rec_id

If (@isDelete=1)
begin

    exec cascade_update 'panel_subtype', @old_value, 'Undefined', 'PANELS', -1
    delete from Corp_Panel_Sub_Type where rec_id = @rec_id
end
else
begin
    if @old_value <> @panel_sub_type_name
    begin
        exec cascade_update 'panel_subtype', @old_value, @panel_sub_type_name, 'PANELS', -1
    end
    update Corp_Panel_Sub_Type
    set panel_sub_type_name = @panel_sub_type_name,
        panel_sub_type_descr = @panel_sub_type_descr where rec_id = @rec_id
end
```



---

Procedure Name: sp\_Edit\_Corp\_Panel\_Test\_Type

```
@panel_test_type_name varchar(50),  
@panel_test_type_descr varchar(250),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

```
declare @old_value varchar(50)  
select @old_value = panel_test_type_name  
From corp_panel_test_type  
where rec_id = @rec_id
```

```
If (@isDelete=1)  
begin
```

```
    exec cascade_update 'panel_testtype', @old_value, 'Undefined',"PANELS",-1  
    delete from Corp_Panel_Test_Type where rec_id = @rec_id
```

```
end
```

```
else
```

```
begin
```

```
    if @old_value <> @panel_test_type_name
```

```
        begin
```

```
            exec cascade_update 'panel_testtype', @old_value, @panel_test_type_name,"PANELS",-1
```

```
        end
```

```
        update Corp_Panel_Test_Type
```

```
        set panel_test_type_name = @panel_test_type_name,
```

```
        panel_test_type_descr = @panel_test_type_descr where rec_id = @rec_id
```

```
    end
```

**Procedure Name: sp\_Edit\_Corp\_Panel\_Type**

```
@panel_type_name varchar(50),  
@panel_type_descr varchar(250),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

```
declare @old_value varchar(50)
```

```
select @old_value = panel_type_name  
  From corp_panel_type  
 where rec_id = @rec_id
```

If (@isDelete=1)

begin

```
  exec cascade_update 'panel_type', @old_value, 'Undefined', 'PANELS', -1
```

```
  delete from Corp_Panel_Type where rec_id = @rec_id
```

end

else

begin

if @old\_value <> @panel\_type\_name

begin

```
  exec cascade_update 'panel_type', @old_value, @panel_type_name, 'PANELS', -1
```

```
  -- exec corp_cascade_update 'panel_type', @old_value, @panel_type_name
```

end

update Corp\_Panel\_Type

```
  set panel_type_name = @panel_type_name,
```

```
  panel_type_descr = @panel_type_descr where rec_id = @rec_id
```

end

Procedure Name: sp\_Edit\_Key

```
@P_KEY_DESCR varchar(250),
@MAIL_QTY_OVERRIDE varchar(10),  --decimal(18,0),
@Newstand_Sale numeric(18,0),
@Newstand_Sale_Marker char(1),
@list_cat_name varchar(250),
@list_name varchar(250),
@list_segment_name varchar(250),
@LIST_COST_ACTUAL_CPM numeric(18,0),
@list_cost_basis varchar(25),
@list_cost_factor numeric(9,2),
@merge_purge_qty numeric(9),
@E_SET_NAME varchar(50),
@R_SET_NAME varchar(50),
@panel_pkg varchar(50),
@subtotal_flag varchar(6),
@optional_effort_id Varchar(50),
@rec_id numeric(8,0),
@isDelete numeric(1,0)
```

AS

If (@isDelete=1)

begin

declare

```
@rPRODUCT_ID numeric(8,2),
@rFULL_PROMO_KEY varchar(50)
```

--get pid and key for recycle key routine

declare c\_key\_info\_r cursor for

select product\_id, full\_promo\_key

from keys where rec\_id = @rec\_id

open c\_key\_info\_r

fetch c\_key\_info\_r into

```
@rPRODUCT_ID,
@rFULL_PROMO_KEY
```

close c\_key\_info\_r

deallocate c\_key\_info\_r

execute doRecycleKey @rPRODUCT\_ID, @rFULL\_PROMO\_KEY

delete from keys where rec\_id = @rec\_id

end

else

begin

--validate data

If not(@MAIL\_QTY\_OVERRIDE > 0)

select @MAIL\_QTY\_OVERRIDE=null

If (@E\_SET\_NAME = "0")

select @E\_SET\_NAME=null

If (@R\_SET\_NAME = "0")

select @R\_SET\_NAME=null

--end data validation

update keys

```
set P_KEY_DESCR=@P_KEY_DESCR,  
MAIL_QTY_OVERRIDE=@MAIL_QTY_OVERRIDE,  
Newstand_Sale = @Newstand_Sale,  
Newstand_Sale_Marker = @Newstand_Sale_Marker,  
LIST_CAT_NAME=isnull(@list_cat_name,'Undefined'),  
LIST_NAME=isnull(@list_name,'Undefined'),  
LIST_SEGMENT_NAME=isnull(@list_segment_name,'Undefined'),  
LIST_COST_ACTUAL_CPM=@LIST_COST_ACTUAL_CPM,  
LIST_COST_BASIS = @list_cost_basis,  
list_cost_factor = @list_cost_factor,  
merge_purge_qty = @merge_purge_qty,  
e_set_name=isnull(@e_set_name,'Undefined'),  
r_set_name=isnull(@r_set_name,'Undefined'),  
PANEL_PKG=isnull(@panel_pkg,'Undefined'),  
subtotal_flag=@subtotal_flag,  
optional_effort_id = @optional_effort_id,  
date_assigned = getdate()  
where rec_id = @rec_id
```

end





## Procedure Name: sp\_Edit\_Panel

```
@panel_name varchar(250), -- 0
@panel_date_start datetime, -- 1
@panel_type varchar(50), -- 2
@panel_subtype varchar(50), -- 3
@test_type varchar(50), -- 4
@panel_pkg varchar(50), -- 5
@e_set_name varchar(50), -- 6
@r_set_name varchar(50), -- 7
--@newstand_sale_marker char(1), -- 8
@login Varchar(50), --9

@rec_id numeric(8,0), -- 1
@isDelete int -- 1

AS
declare
    @IPRODUCT_ID numeric(8,0),
    @IPANEL_NAME varchar(50),
    @ICAMP_NAME varchar(50),
    @ISOURCE_NAME varchar(250),
    @ISRC_CAT_NAME varchar(30),
    @lFull_promo_key varchar(25),
    @lkeys_product_id numeric(8,0),
    @lE_SET_NAME varchar(50),
    @lR_SET_NAME varchar(50),
    @lpanel_pkg varchar(50),
    @message Varchar(255),
    @src_indicator char(3),
    @grp_name varchar(50),
    @copies int,
    @copypanelname varchar(50)

--get panel and other info for deletions from key table
select @IPRODUCT_ID = PRODUCT_ID,
    @IPANEL_NAME = PANEL_NAME,
    @ICAMP_NAME = CAMP_NAME,
    @ISOURCE_NAME = SOURCE_NAME,
    @ISRC_CAT_NAME = SRC_CAT_NAME,
    @le_set_name = e_set_name,
    @lr_set_name = r_set_name,
    @lpanel_pkg = panel_pkg,
    @src_indicator = src_indicator,
    @grp_name = grp_name

from panels
where rec_id = @rec_id

If (@isDelete=1)
begin

    declare c_keys cursor for /* r101 recordset - the main/reference one */
```

```
select product_id, full_promo_key
from keys
where product_id = @lproduct_id
  and camp_name = @lcamp_name
  and SOURCE_NAME = @lSOURCE_NAME
  and panel_name = @lpanel_name
  and src_cat_name = @lsrc_cat_name

open c_keys
fetch c_keys into @lkeys_product_id, @lfull_promo_key

while (@@fetch_status = 0)
  begin
    exec doRecycleKey @lkeys_product_id, @lfull_promo_key
    fetch c_keys into @lkeys_product_id, @lfull_promo_key
  end
close c_keys
deallocate c_keys

delete from keys
where product_id = @lPRODUCT_ID
  and src_cat_name = @lSRC_CAT_NAME
  and source_name = @lSOURCE_NAME
  and camp_name = @lCamp_name
  and panel_name = @lPANEL_NAME

delete from panels
where rec_id = @rec_id

--close c_panel_info2
--deallocate c_panel_info2
set @message = 'Panel:' + @lPANEL_NAME + ',Campaign:' + @lCAMP_NAME + ',Source:' +
@lSOURCE_NAME + ',Src_cat:' + @lsrc_cat_name + ',delete applied to panel and keys'
exec sp_loginfo @lproduct_id, @login, 'Delete Panel', @message

end

if (@IsDelete=0)
begin
--declare
--get expense and revenue set names for pass down if blank
-- NOT USING ANYMORE, this stuff should be defined when keys are assigned
--      select @lE_Set_name = E_SET_NAME,
--      @lR_Set_name = R_SET_NAME
-- from campaigns where product_id = @lPRODUCT_ID
--and SRC_CAT_NAME = @lSRC_CAT_NAME and SOURCE_NAME = @lSOURCE_NAME and
CAMP_NAME = @lCAMP_NAME

PRINT 'e set name--' + @e_set_name + '--e set name --' + @lE_SET_NAME + '--' + CAST(
@lPRODUCT_ID as varchar)

update keys
  set panel_name = @panel_name,
  panel_type=isnull(@panel_type,'Undefined'),
  panel_subtype=isnull(@panel_subtype,'Undefined'),
```

```

test_type=isnull(@test_type,'Undefined'),
-- panel_pkg=isnull(@panel_pkg,'Undefined'),
panel_date_start=@panel_date_start
-- newstand_sale_marker = @newstand_sale_marker
-- e_set_name=isnull(@e_set_name,@lE_SET_NAME),
-- r_set_name=isnull(@r_set_name,@lR_SET_NAME)
where (panel_name + ':' + camp_name + ':' + src_cat_name + ':' + source_name + ':' +
cast(product_id as varchar) )
= (select panel_name + ':' + camp_name + ':' + src_cat_name + ':' + source_name + ':' +
cast(product_id as varchar) from panels where rec_id = @rec_id )
if (@le_set_name <> @e_set_name)
begin
update keys
set e_set_name = @e_set_name
-- panel_type=isnull(@panel_type,'Undefined'),
-- panel_subtype=isnull(@panel_subtype,'Undefined'),
-- test_type=isnull(@test_type,'Undefined'),
-- panel_offer=@panel_offer,
-- panel_offer_price=@panel_offer_price,
-- panel_pkg=isnull(@panel_pkg,'Undefined'),
-- panel_premium=@panel_premium,
-- panel_date_start=@panel_date_start,
-- panel_date_end=@panel_date_end

-- r_set_name=isnull(@r_set_name,@lR_SET_NAME)
where e_set_name = @le_set_name and
(panel_name + ':' + camp_name + ':' + src_cat_name + ':' + cast(product_id as varchar) )
= (select panel_name + ':' + camp_name + ':' + src_cat_name + ':' + cast(product_id as
varchar)
from panels where rec_id = @rec_id )
end
if (@lr_set_name <> @r_set_name)
begin
update keys
set r_set_name = @r_set_name
-- panel_type=isnull(@panel_type,'Undefined'),
-- panel_subtype=isnull(@panel_subtype,'Undefined'),
-- test_type=isnull(@test_type,'Undefined'),
-- panel_offer=@panel_offer,
-- panel_offer_price=@panel_offer_price,
-- panel_pkg=isnull(@panel_pkg,'Undefined'),
-- panel_premium=@panel_premium,
-- panel_date_start=@panel_date_start,
-- panel_date_end=@panel_date_end

-- r_set_name=isnull(@r_set_name,@lR_SET_NAME)
where r_set_name = @lr_set_name and
(panel_name + ':' + camp_name + ':' + src_cat_name + ':' +
source_name + ':' + cast(product_id as varchar) ) =
(select panel_name + ':' + camp_name + ':' + src_cat_name + ':' +
source_name + ':' + cast(product_id as varchar)
from panels where rec_id = @rec_id )
end
if (@lPanel_pkg <> @panel_pkg)
begin

```



any one of the things that we have  
done in the past is to have a  
policy of non-interference in the  
internal affairs of other countries  
and to have a policy of non-alignment  
with any one of the great powers  
of the world. This policy has been  
the basis of our foreign policy for  
many years and it has been the  
basis of our success in the world.

```

update keys
    set Panel_Pkg = @panel_pkg
-- panel_type=isnull(@panel_type,'Undefined'),
-- panel_subtype=isnull(@panel_subtype,'Undefined'),
-- test_type=isnull(@test_type,'Undefined'),
-- panel_offer=@panel_offer,
-- panel_offer_price=@panel_offer_price,
-- panel_pkg=isnull(@panel_pkg,'Undefined'),
-- panel_premium=@panel_premium,
-- panel_date_start=@panel_date_start,
-- panel_date_end=@panel_date_end

-- r_set_name=isnull(@r_set_name,@IR_SET_NAME)
--   where Panel_Pkg = @lpanel_pkg and
--         (panel_name + ':' + camp_name + ':' + src_cat_name + ':' +
--          source_name + ':' + cast(product_id as varchar) ) =
--         (select panel_name + ':' + camp_name + ':' + src_cat_name + ':' +
--          source_name + ':' + cast(product_id as varchar)
--          from panels where rec_id = @rec_id )
end

update panels
    set panel_name = @panel_name,
    panel_type=isnull(@panel_type,'Undefined'),
    panel_subtype=isnull(@panel_subtype,'Undefined'),
    test_type=isnull(@test_type,'Undefined'),
    panel_pkg=isnull(@panel_pkg,'Undefined'),
    panel_date_start=@panel_date_start,
    e_set_name=isnull(@e_set_name,@IE_SET_NAME),
    r_set_name=isnull(@r_set_name,@IR_SET_NAME)
-- newstand_sale_marker = @newstand_sale_marker
-- where rec_id = @rec_id

if @lpanel_name <> @panel_name
BEGIN
    set @message = 'Old Panel:' + @IPANEL_NAME + ',New Panel:' + @Panel_name +
    ',Campaign:' + @ICAMP_NAME + ',Source:' + @ISOURCE_NAME + ',Src_cat:' +
    @lsrc_cat_name + ',edit applied to panel and keys'

END
ELSE
BEGIN

    set @message = 'Panel:' + @PANEL_NAME +
    ',Campaign:' + @ICAMP_NAME + ',Source:' + @ISOURCE_NAME + ',Src_cat:' +
    @lsrc_cat_name + ',edit applied to panel and keys'
end
exec sp_loginfo @lproduct_id, @login, 'Edit Panel', @message

end

if (@IsDelete=2) --then copy existing panel

```



begin

```
set @copypanelname = 'Copy __ of' + @panel_name
--Get number of next copy
select @copies = (count(*) + 1)
    from panels
    where panel_name like @copypanelname
    and camp_name = @lCamp_name
    and src_cat_name = @lsrc_cat_name
    and product_id = @lproduct_id
    and source_name = @lsource_name
if @copies < 10 set @copypanelname = "Copy " + cast(@copies as varchar(1)) + " of "
else          set @copypanelname = "Copy " + cast(@copies as varchar(2)) + " of "
set @panel_name = @copypanelname + @panel_name
insert into panels (panel_name
    , panel_type
    , panel_subtype
    , test_type
    , panel_pkg
    , panel_date_start
    , e_set_name
    , r_set_name
    , product_id
    , camp_name
    , source_name
    , src_cat_name
    , src_indicator
    , grp_name)
values
    (@panel_name
    , isnull(@panel_type,'Undefined')
    , isnull(@panel_subtype,'Undefined')
    , isnull(@test_type,'Undefined')
    , isnull(@panel_pkg,'Undefined')
    , @panel_date_start
    , isnull(@e_set_name,@IE_SET_NAME)
    , isnull(@r_set_name,@IR_SET_NAME)
    , @IPRODUCT_ID
    , @ICAMP_NAME
    , @ISOURCE_NAME
    , @ISRC_CAT_NAME
    , @src_indicator
    , @grp_name
    )
```

end

---

Procedure Name: sp\_edit\_prior\_Source\_desc

```
@prior_source_identifier char (2)
, @prior_source_desc varchar (20)
, @billing_cur_src_desc VARCHAR(255)
, @model_source_cat varchar(50)
, @model_source varchar(50)
, @agent_src_desc varchar(50)
, @valid_from_date datetime
, @valid_to_date datetime
, @rec_id numeric(8,0)
, @isDelete int

AS
declare @old_value varchar(50), @old_product_id numeric(9)

select @old_value = prior_source_identifier,
       @old_product_id = product_id
  From prior_source_desc
 where rec_id = @rec_id

If (@isDelete=1)
    begin
        exec cascade_update 'prior_source_identifier', @old_value, '-', 'KEYS', @old_product_id
        delete from prior_source_desc where rec_id = @rec_id
    end
else
    begin
        exec cascade_update 'prior_source_identifier', @old_value, '-', 'Est_max_effort',
        @old_product_id
        update prior_source_desc
           set --prior_source_identifier = @prior_source_identifier ,
               prior_source_desc = @prior_source_desc
             , billing_cur_src_desc = @billing_cur_src_desc
             , valid_from_date = @valid_from_date
             , valid_to_date = @valid_to_date
             , agent_src_desc = @agent_src_desc
             , model_source = @model_source
             , model_source_cat = @model_source_cat
           where rec_id = @rec_id
    end

end
```



Procedure Name: sp\_Edit\_Prod\_Camp\_Status

```
@camp_status_name varchar(20),  
@camp_status_descr varchar(250),
```

```
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

Declare @old\_value VARCHAR(255), @product\_id numeric(9)

```
select @old_value =  
    camp_status_name,  
    @product_id = product_id  
FROM prod_camp_status  
WHERE rec_id = @rec_id
```

If (@isDelete=1)  
begin

```
    exec cascade_update 'camp_status_name', @old_value, 'Undefined', 'CAMPAIGNS', @product_id  
    delete from Prod_Camp_Status where rec_id = @rec_id
```

end  
else

begin

```
    if @old_value <> @camp_status_name
```

```
        begin
```

```
            exec cascade_update 'camp_status_name', @old_value, @camp_status_name, 'CAMPAIGNS',
```

```
@product_id
```

```
        -- exec prod_cascade_update_w_camp 'camp_status_name', @old_value, @camp_status_name ,
```

```
@product_id
```

```
        end
```

update Prod\_Camp\_Status

```
    set camp_status_name = @camp_status_name,
```

```
    camp_status_descr = @camp_status_descr where rec_id = @rec_id
```

end





Procedure Name: sp\_Edit\_Prod\_Camp\_Type

```
@camp_type_name varchar(20),  
@camp_type_descr varchar(250),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

Declare @old\_value VARCHAR(255), @product\_id numeric(9)

```
select @old_value =  
    camp_type_name,  
    @product_id = product_id  
FROM prod_camp_type  
WHERE rec_id = @rec_id
```

If (@isDelete=1)

begin

```
    exec cascade_update 'camp_type', @old_value, 'Undefined', 'CAMPAIGNS', @product_id  
    -- exec prod_cascade_update_w_camp 'camp_type', @old_value, 'Undefined', @product_id
```

```
    delete from Prod_Camp_Type where rec_id = @rec_id
```

end

else

begin

```
    if @old_value <> @camp_type_name
```

begin

```
        exec cascade_update 'camp_type_name', @old_value, @camp_type_name, 'CAMPAIGNS',
```

```
@product_id
```

```
--        exec prod_cascade_update_w_camp 'camp_type', @old_value, @camp_type_name, @product_id
```

end

update Prod\_Camp\_Type

```
    set camp_type_name = @camp_type_name,
```

```
    camp_type_descr = @camp_type_descr where rec_id = @rec_id
```

end



Procedure Name: sp\_Edit\_Prod\_E\_Set

```
@product_id numeric(8, 0) ,
@e_set_name varchar (50) ,
@print_cost numeric(11, 3) ,
@print_cpm_cpu varchar (10) ,
@print_basis varchar (20) ,
@merge_purge numeric(11, 3) ,
@merge_cpm_cpu varchar (10) ,
@merge_basis varchar (20) ,
@ltr_cost numeric(11, 3) ,
@ltr_cpm_cpu varchar (10) ,
@ltr_basis varchar (20) ,
@postage_out_cost numeric(11, 3) ,
@postage_out_cpm_cpu varchar (10) ,
@postage_out_basis varchar (20) ,
@postage_in_cost numeric(11, 3) ,
@postage_in_cpm_cpu varchar (10) ,
@postage_in_basis varchar (20) ,
@premium_cost numeric(11, 3) ,
@premium_cpm_cpu varchar (10) ,
@premium_basis varchar (20) ,
@badpay_cost numeric(11, 3) ,
@badpay_cpm_cpu varchar (10) ,
@badpay_basis varchar (20) ,
@billing_cost numeric(11, 3) ,
@billing_cpm_cpu varchar (10) ,
@billing_basis varchar (20) ,
@subs_svc_costs numeric(11, 3) ,
@subs_svc_cpm_cpu varchar (10) ,
@subs_svc_basis varchar (20) ,
@exp_other_costs numeric(11, 3) ,
@exp_other_cpm_cpu varchar (10) ,
@exp_other_basis varchar (20) ,
@rec_id numeric(8,0),
@isDelete numeric(1,0)
```

AS

```
Declare @old_value VARCHAR(255), @my_product_id numeric(9)
declare @e_set_id numeric(9)
```

```
select @old_value =
    e_set_name,
    @my_product_id = product_id
FROM expense_sets
WHERE rec_id = @rec_id
```

If (@isDelete=1)

begin

```
exec cascade_update 'e_set_name', @old_value, 'Undefined','CAMPAIGNS', @product_id
delete from Expense_sets where rec_id = @rec_id
```

end

```

else
begin
  if @old_value <> @e_set_name
  begin
    set @e_set_id = -1
    select @e_set_id = rec_id
      from expense_sets
     where e_set_name = @e_set_name
       and product_id = @product_id

    if @e_set_id = -1 exec sp_Add_Prod_E_Set --Create New Record, set everything to new e set and
delete old one
      @product_id, @e_set_name, @print_cost, @print_cpm_cpu, @print_basis,
@merge_purge, @merge_cpm_cpu, @merge_basis
      , @ltr_cost, @ltr_cpm_cpu, @ltr_basis, @postage_out_cost, @postage_out_cpm_cpu,
@postage_out_basis, @postage_in_cost, @postage_in_cpm_cpu
      , @postage_in_basis, @premium_cost, @premium_cpm_cpu, @premium_basis,
@badpay_cost, @badpay_cpm_cpu, @badpay_basis, @billing_cost
      , @billing_cpm_cpu, @billing_basis, @subs_svc_costs, @subs_svc_cpm_cpu,
@subs_svc_basis, @exp_other_costs, @exp_other_cpm_cpu
      , @exp_other_basis

    exec cascade_update 'e_set_name', @old_value, @e_set_name, 'CAMPAIGNS', @product_id
    delete from expense_sets where rec_id = @rec_id
  end
else
  update expense_sets
    set e_set_name = @e_set_name ,
        print_cost = @print_cost ,
        print_cpm_cpu = @print_cpm_cpu ,
        print_basis = @print_basis ,
        merge_purge = @merge_purge ,
        merge_cpm_cpu = @merge_cpm_cpu ,
        merge_basis = @merge_basis ,
        ltr_cost = @ltr_cost ,
        ltr_cpm_cpu = @ltr_cpm_cpu ,
        ltr_basis = @ltr_basis ,
        postage_out_cost = @postage_out_cost ,
        postage_out_cpm_cpu = @postage_out_cpm_cpu ,
        postage_out_basis = @postage_out_basis ,
        postage_in_cost = @postage_in_cost ,
        postage_in_cpm_cpu = @postage_in_cpm_cpu ,
        postage_in_basis = @postage_in_basis ,
        premium_cost = @premium_cost ,
        premium_cpm_cpu = @premium_cpm_cpu ,
        premium_basis = @premium_basis ,
        badpay_cost = @badpay_cost ,
        badpay_cpm_cpu = @badpay_cpm_cpu ,
        badpay_basis = @badpay_basis ,
        billing_cost = @billing_cost ,
        billing_cpm_cpu = @billing_cpm_cpu ,
        billing_basis = @billing_basis ,
        subs_svc_costs = @subs_svc_costs ,
        subs_svc_cpm_cpu = @subs_svc_cpm_cpu ,
        subs_svc_basis = @subs_svc_basis ,

```

```
exp_other_costs = @exp_other_costs ,  
exp_other_cpm_cpu = @exp_other_cpm_cpu ,  
exp_other_basis = @exp_other_basis  
where rec_id = @rec_id  
end
```



Procedure Name: sp\_Edit\_Prod\_List\_Cat

```
    @name varchar(250),
    @descr varchar(250),
    @rec_id numeric(8,0),
    @isDelete numeric(1,0)
AS
Declare @old_value VARCHAR(255),
        @product_id numeric(9)

select @old_value =
        list_cat_name,
        @product_id = product_id
FROM list_cats
WHERE rec_id = @rec_id

If (@isDelete=1)
begin
    exec cascade_update 'LIST_CAT_NAME', @old_value, 'Undefined', "KEYS", @product_id
    delete from List_Cats where rec_id = @rec_id
end
else
begin
    if @name <> @old_value
    begin
        exec cascade_update 'LIST_CAT_NAME', @old_value, @name, "KEYS", @product_id
    end
end

update List_Cats
    set List_Cat_name = @name,
        List_Cat_descr = @descr where rec_id = @rec_id
end
```



Procedure Name: sp\_Edit\_Prod\_List\_Name

```
@name varchar(250),  
@descr varchar(250),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

Declare @old\_value VARCHAR(255), @product\_id numeric(9)

```
select @old_value =  
    list_name,  
    @product_id = product_id  
FROM list_names  
WHERE rec_id = @rec_id
```

If (@isDelete=1)

begin

```
exec cascade_update 'LIST_NAME', @old_value, 'Undefined', "KEYS", @product_id  
delete from List_Names where rec_id = @rec_id
```

end

else

begin

if @name <> @old\_value

begin

```
exec cascade_update 'LIST_NAME', @old_value, @name, "KEYS", @product_id
```

end

update List\_NAMES

```
set List_name = @name,
```

```
List_Name_description = @descr
```

```
where rec_id = @rec_id
```

end



Procedure Name: sp\_Edit\_Prod\_List\_Segment

```
@name varchar(250),
@descr varchar(250),
@rec_id numeric(8,0),
@isDelete numeric(1,0)
AS

Declare @old_value VARCHAR(255),
        @product_id numeric(9)

select @old_value =
        list_segment_name,
        @product_id = product_id
FROM list_segments
WHERE rec_id = @rec_id

If (@isDelete=1)
begin
    exec cascade_update 'LIST_SEGMENT_NAME', @old_value, 'Undefined', "KEYS", @product_id
    delete from List_Segments where rec_id = @rec_id
end
else
begin
    if @name <> @old_value
    begin
        exec cascade_update 'LIST_SEGMENT_NAME', @old_value, @name, "KEYS", @product_id
    end

    update List_Segments
        set List_Segment_name = @name,
            List_Segment_description = @descr where rec_id = @rec_id
end
```

Procedure Name: sp\_Edit\_Prod\_Panel\_Package

```
@package_name varchar(50),  
@package_description varchar(250),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

```
If (@isDelete=1)  
begin  
delete from prod_panel_package where rec_id = @rec_id  
end  
else  
begin  
update prod_panel_package  
set package_name = @package_name,  
package_description = @package_description where rec_id = @rec_id  
end
```





Procedure Name: sp\_edit\_Prod\_R\_Set

```
@product_id numeric(8, 0),
@r_set_name varchar (50) ,
@subs_revenue numeric(11, 3) ,
@subs_cpm_cpu varchar (10) ,
@ad_revenue numeric(11, 3),
@ad_cpm_cpu varchar (10),
@listrental_revenue numeric(11, 3) ,
@listrental_cpm_cpu varchar (10),
@other_revenue numeric(11, 3),
@other_cpm_cpu varchar (10),
@rec_id numeric(8,0),
@isDelete numeric(1,0)
```

AS

Declare @old\_value VARCHAR(255), @my\_product\_id numeric(9)

```
select @old_value = r_set_name,
       @my_product_id = product_id
FROM revenue_sets
WHERE rec_id = @rec_id
```

If (@isDelete=1)

begin

```
exec cascade_update 'r_set_name', @old_value, 'Undefined','CAMPAIGNS', @product_id
delete from Revenue_sets where rec_id = @rec_id
```

end

else

begin

```
if @old_value <> @r_set_name
```

begin

```
exec cascade_update 'r_set_name', @old_value, @r_set_name, 'CAMPAIGNS', @product_id
```

end

update revenue\_sets

```
set product_id = @product_id ,
  r_set_name = @r_set_name ,
  subs_revenue = @subs_revenue ,
  subs_cpm_cpu = @subs_cpm_cpu ,
  ad_revenue = @ad_revenue,
  ad_cpm_cpu = @ad_cpm_cpu,
  listrental_revenue= @listrental_revenue,
  listrental_cpm_cpu = @listrental_cpm_cpu ,
  other_revenue = @other_revenue,
  other_cpm_cpu = @other_cpm_cpu
```

```
where rec_id = @rec_id
```

end



Procedure Name: sp\_Edit\_Source

```
@source_name varchar(250),  
@src_indicator varchar(10),  
@login Varchar(50),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

If (@isDelete=1)

begin

delete from source where rec\_id = @rec\_id

end

else

begin

declare

```
@PRODUCT_ID numeric(8,0),  
@src_cat_name varchar(50),  
@old_source_name varchar(30),  
@old_src_ind varchar(3),  
@message VARCHAR(200)
```

```
SELECT @PRODUCT_ID = Product_id ,  
@src_cat_name = src_cat_name,  
@old_source_name = source_name ,  
@old_src_ind = src_indicator
```

FROM source where rec\_id = @rec\_id

IF @source\_name <> @old\_source\_name or @src\_indicator <> @old\_src\_ind

BEGIN

set @message = 'Change applied to Panel, Campaign and Key levels as well as Source'

UPDATE campaigns

SET source\_name = @source\_name,

src\_indicator = @src\_indicator

WHERE Product\_id = @Product\_id

and src\_cat\_name = @src\_cat\_name

and source\_name = @old\_source\_name

and src\_indicator = @old\_src\_ind

UPDATE Panels

SET source\_name = @source\_name,

src\_indicator = @src\_indicator

WHERE Product\_id = @Product\_id

and src\_cat\_name = @src\_cat\_name

and source\_name = @old\_source\_name

and src\_indicator = @old\_src\_ind

UPDATE keys

SET source\_name = @source\_name,

src\_indicator = @src\_indicator

WHERE Product\_id = @Product\_id

and src\_cat\_name = @src\_cat\_name

```
and source_name = @old_source_name  
and src_indicator = @old_src_ind
```

```
update source
```

```
  set source_name = @source_name,  
      src_indicator = @src_indicator where rec_id = @rec_id
```

```
exec sp_logininfo @product_id, @login, 'Edit Source', @message  
end  
end
```



Procedure Name: sp\_edit\_Source\_description

```
@u_source_name varchar(250),  
@u_source_description varchar(250),  
@rec_id numeric(8,0),  
@isDelete numeric(1,0)
```

AS

```
If (@isDelete=1)  
begin  
delete from Source_descriptions where rec_id = @rec_id  
end  
else  
begin  
update Source_descriptions  
set u_source_name = @u_source_name,  
u_source_description = @u_source_description where rec_id = @rec_id  
end
```



Procedure Name: sp\_LogInfo

```
@product_id varchar(250),  
@login varchar(50),  
--log_timestamp  
@action varchar(50),  
@action_description varchar(100)
```

As

```
insert into logs (product_id,login,log_timestamp,action,action_description) values  
(@product_id,@login,getdate(),@action,@action_description)
```



Procedure Name: sp\_RemoveDuplicate101Keys

@pid numeric(9)

AS

declare @r101\_key varchar(10), @select\_date smalldatetime, @product\_id numeric(8,0), @rct\_rec\_num decimal(9), @r101\_reorg\_date smalldatetime, @keyCount numeric(8,0)

declare c\_r101dup cursor for

select product\_id, r101\_key, r101\_reorg\_date, count(r101\_key) as cnt, max(rct\_rec\_num) as maxRecNum  
from r101\_flow where product\_id = @pid group by product\_id, r101\_key, r101\_reorg\_date order by  
product\_id, cnt desc, r101\_key

open c\_r101dup

fetch c\_r101dup into @product\_id, @r101\_key, @r101\_reorg\_date, @keyCount, @rct\_rec\_num

while (@@fetch\_status = 0 and @KeyCount > 1)

begin  
delete from r101\_flow where product\_id = @product\_id and r101\_reorg\_date =  
@r101\_reorg\_date and r101\_key = @r101\_key and rct\_rec\_num < @rct\_rec\_num  
fetch c\_r101dup into @product\_id, @r101\_key, @r101\_reorg\_date, @keyCount, @rct\_rec\_num  
end

close c\_r101dup

deallocate c\_r101dup



Procedure Name: sp\_RemoveDuplicate401Keys

@pid numeric(9)

AS

declare @r401\_key varchar(10), @select\_date smalldatetime, @product\_id numeric(8,0), @r401\_rec\_num decimal(9), @r401\_reorg\_date smalldatetime, @keyCount numeric(8,0)

declare c\_r401dup cursor for

select product\_id, r401\_key, r401\_reorg\_date, count(r401\_key) as cnt, max(r401\_rec\_num) as maxRecNum  
from r401\_flow where product\_id = @pid group by product\_id, r401\_key, r401\_reorg\_date order by  
product\_id, cnt desc, r401\_key

open c\_r401dup

fetch c\_r401dup into @product\_id, @r401\_key, @r401\_reorg\_date, @keyCount, @r401\_rec\_num

while (@@fetch\_status = 0 and @KeyCount > 1)

begin

delete from r401\_flow where product\_id = @product\_id and r401\_reorg\_date =

@r401\_reorg\_date and r401\_key = @r401\_key and r401\_rec\_num < @r401\_rec\_num

fetch c\_r401dup into @product\_id, @r401\_key, @r401\_reorg\_date, @keyCount, @r401\_rec\_num

end

close c\_r401dup

deallocate c\_r401dup



Procedure Name: sp\_UpdateCamp\_Statues

```
@product_id numeric(9),  
@camp_status_name varchar(20),  
@camp_status_descr varchar(250),  
@actionParam varchar(20),  
@kname varchar(20)
```

AS

```
If (@actionParam='AddRec')  
begin  
insert into camp_statuses (product_id, camp_status_name, camp_status_descr) values (@product_id,  
@camp_status_name, @camp_status_descr)  
end  
else  
If (@actionParam='EditRec')  
begin  
update camp_statuses set camp_status_name = @camp_status_name, camp_status_descr =  
@camp_status_descr  
where product_id = @product_id and camp_status_name = @kName  
end
```



Procedure Name: sp\_UpdateCamp\_Types

```
@product_id numeric(9),  
@camp_type varchar(50),  
@camp_type_descr varchar(250),  
@actionParam varchar(20),  
@kname varchar(20)
```

AS

```
If (@actionParam='AddRec')  
begin  
insert into camp_types (product_id, camp_type, camp_type_descr) values (@product_id, @camp_type,  
@camp_type_descr)  
end  
else  
If (@actionParam='EditRec')  
begin  
update camp_types set camp_type = @camp_type, camp_type_descr = @camp_type_descr  
where product_id = @product_id and camp_type = @kName  
end
```



Procedure Name: sp\_UpdatePanel\_SubTypes

```
@product_id numeric(9),  
@panel_subtype varchar(50),  
@panel_subtype_descr varchar(250),  
@actionParam varchar(20),  
@kname varchar(20)
```

AS

```
If (@actionParam='AddRec')  
begin  
insert into p_sub_types (product_id, panel_subtype, panel_subtype_descr) values (@product_id,  
@panel_subtype, @panel_subtype_descr)  
end  
else  
If (@actionParam='EditRec')  
begin  
update p_sub_types set panel_subtype = @panel_subtype, panel_subtype_descr = @panel_subtype_descr  
where product_id = @product_id and panel_subtype = @kName  
end
```

Procedure Name: sp\_UpdatePanel\_TestTypes

```
@product_id numeric(9),  
@test_type varchar(50),  
@test_type_descr varchar(250),  
@actionParam varchar(20),  
@kname varchar(20)
```

AS

```
If (@actionParam='AddRec')  
begin  
insert into p_test_types (product_id, test_type, test_type_descr) values (@product_id, @test_type,  
@test_type_descr)  
end  
else  
If (@actionParam='EditRec')  
begin  
update p_test_types set test_type = @test_type, test_type_descr = @test_type_descr  
where product_id = @product_id and test_type = @kName  
end
```



**Procedure Name: sp\_UpdatePanel\_Types**

```
@product_id numeric(9),  
@panel_type varchar(50),  
@panel_type_descr varchar(250),  
@actionParam varchar(20),  
@kname varchar(20)
```

AS

```
If (@actionParam='AddRec')  
begin  
insert into p_types (product_id, panel_type, panel_type_descr) values (@product_id, @panel_type,  
@panel_type_descr)  
end  
else  
If (@actionParam='EditRec')  
begin  
update p_types set panel_type = @panel_type, panel_type_descr = @panel_type_descr  
where product_id = @product_id and panel_type = @kName  
end
```



Procedure Name: spGetRPTList

```
@src_cat_name varchar(30),  
@src_sub_cat varchar(50)
```

AS

```
SELECT report_id,rpt_name, rpt_description, location, rpt_filters  
FROM rpt_cats  
WHERE src_cat_name = @src_cat_name  
AND src_sub_cat = @src_sub_cat
```



## CONTENTS

---

### IMT REPORTING

About IMT	04
Overview	
System Requirements	05

---

### ADMINISTRATION

Overview	06
Access	07

---

### IMT Tool Bar Features

Fast Find	08
Search	08
Available Keys	08
Setup Tables Chart	09
Using Setup Tables	10

---

### Build Campaigns

Step 1 – Add Source	11
Step 2 – Add or Copy Campaigns	12
Step 3 – Add or Copy Panels	13
Step 4 – Assign Keys	14

---

### Group Edit and Multi-Edit

Overview	15
----------	----

---

### Edit Campaigns

Step 1 – Edit Campaigns	16
Step 2 – Edit Panels	17
Step 3 – Edit Keys	18

---

### AutoAdmin

Overview	19
Setup	20
How Tables Work Together	21
Source Key Anatomy	22
Create Setup Schemes	24
Create New Source	25
Campaign Name Definition	26
Panel Name Definition	27
Renewal Prior Source Definitions	28

---

### Insert Card Setup

Overview	29
----------	----



## CONTENTS

### REPORTING

How to Start Using Reports	30
Navigating Reports	31
Reading Reports	32

### AGENT

Overview	33
Field Definitions	34
List of Available Reports	35

### BILLING

Overview	36
Field Definitions	37
List of Available Reports	39

### DTP

Overview	40
Field Definitions	41
List of Available Reports	43

### GIFT

Overview	44
Field Definitions	45
List of Available Reports	47

### RENEWAL

Overview	48
Field Definitions	49
List of Available Reports	51

### GLOSSARY

A - C	52
D - L	53
M - P	54
Q - S	55
T - Z	56

### REPORT SAMPLES

Samples	58
---------	----



Integrated Marketing Technology is an application service provider (ASP) founded in 1993 to be the premier provider of integrated marketing consulting and database services to subscription and membership-based marketing companies. IMT's mission is to allow businesses to make faster, more informed, and more strategic decisions through its sophisticated technology tools and marketing services.

### "Information at Your Fingertips"

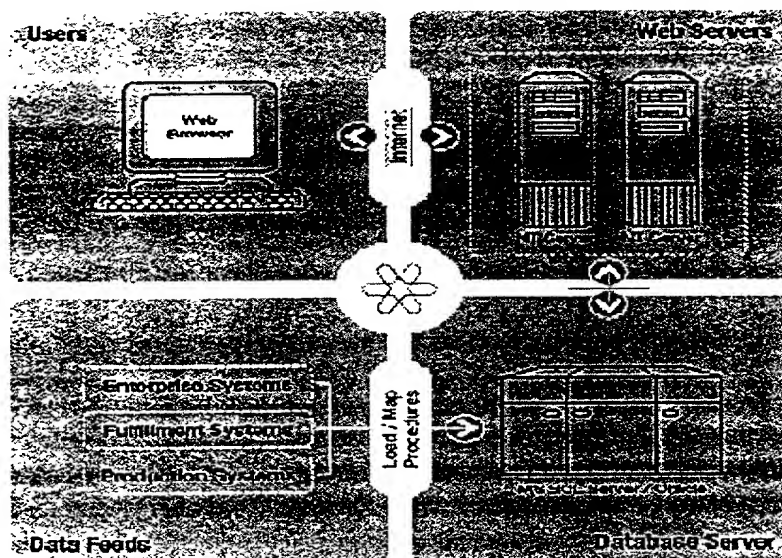
We provide actionable marketing information directly to the people who have their finger on the pulse of the business. It's the only way to turn raw data into knowledge, and it's the key to our company's, and our client's, success. IMT's systems allow instant, real-time access to critical corporate data and analysis – from high-level assessment of sales channels to detailed evaluation of individual campaigns or lists.

A flexible and powerful web-based application requiring no specialized hardware or software, IMT Reporting offers companies accurate and consistent analysis of their marketing initiatives. IMT Reporting provides highly segmented, clearly defined marketing analysis at almost any level – from source channel to campaign to list.

### The Benefits:

- Proven, insightful customer acquisition and retention analysis
- Improved Lifetime customer value reporting
- Channel ranking and analysis based on order volume and profitability
- Multi-contract customer depletion analysis
- Consistent interpretation of data
- Cross-product, cross-organizational reporting
- Full historical marketing information and analysis
- Improved marketing ROI
- Dramatic cost savings

Through IMT's Internet-based reporting products, marketers have more control over when they receive and disperse critical marketing data. Businesses can decide whether they want to receive daily or weekly data feeds from order processing and fulfillment operations systems. Whatever frequency they choose, they will get this information in real time.







### PC Computers

- An Intel® i486™ or Pentium® processor – based computer (Pentium recommended), Microsoft Windows 95, Windows 98, or Windows NT® 4.0 with service Pack 3 or later, 16 MB of RAM on Windows 95 and Windows 98, 24 MB of RAM for Windows NT (32 MB recommended) and Hard drive with at least 75 MB of available space.

### Mac Computers

- Apple Power Macintosh computer, Apple System Software version 8.6 or later, 6 MB of RAM (12 MB recommended) and Hard Drive with at least 75 MB of available space.

### Other Hardware Requirements / Recommendations

- Monitor – Size: 15" or larger (recommended), Color: 256 or higher (i.e. 16 bit), Screen Area: 800 x 600 (recommended 1024 x 768), Keyboard - American Standard 101 key, Pointing Devices - Mouse or Track ball, Local or Networked Laser Printer – Paper Size: 8 1/2 x 11 " + Orientation: Landscape

### Recommended Software

- Email program - communicate with [ClientServices@IMTnetwork.com](mailto:ClientServices@IMTnetwork.com) & [Support@IMTnetwork.com](mailto:Support@IMTnetwork.com)
- Spread Sheet program – export data from IMT Reporting (i.e. charts)

### Internet Connection

- Via Modem: Dial-up 56K (minimum)
- Via DSL / ISDN / T-1 = Local Area Network (recommended)

### Browser Check and Installation Internet Explorer (IE)

- IMT Reporting is best viewed through Internet Explorer 5.0, available for both PCs and Macintosh computers. The system requirement for loading Explorer 5.0 is as follows:
- **NOTE:** I.E.4.0, will work with basic features

#### Step 1 – Check your system for Internet Explorer 5.0, confirm the version you're running.

- Start Explorer
- Click on **Help** located on the menu bar
- Click on **About Internet Explorer** and a new window will open
- Verify the version number.
- If the version number is 5.0 or later, then go to Step 3.

**Step 2 – If the version number is less than 5.0, or if you do not have Explorer installed on your computer, then go to Microsoft's web site and follow the upgrade or installation instructions, or contact your IT department to have Internet Explorer 5.0 installed.**

< <http://www.microsoft.com/windows/ie/default.htm> >

#### Step 3 – Configuring Internet Explorer 5.0

- Start Explorer
- Click on **Tools** located on the menu bar
- Click on **Internet Options** and a new window will open
- Click on **Settings for Temporary Internet Files** located on the **General Tab**
- Click on **Every visit to page** located below **Check for newer versions of stored pages**
- Click on **OK**
- Click on **Apply** then **OK** on the **Internet Options** window
- Test your access by going to < <http://www.IMTnetwork.com/> > and logging onto the system by clicking **Client Center**, and enter username and password (case sensitive) to successfully log onto the system.



**IMT Reporting** has two main components: **Admin** and **Reporting**. **Admin** generally refers to the administration processes, such as assigning promotion keys to specific **sources**, **campaigns** and **panels**.

#### What You Need to Get Started?

- Authorize IMT to accept data feeds weekly from your fulfillment house (FTP recommended).
- Gather all documentation and materials normally required to prepare detailed reports:
  - **Source Names** (source ID)
  - **Campaign:** Names, Start Dates, Types and Expense detail
  - **Panel:** Names, Start Dates, Test Types and Package Types
  - **Key:** Mail Qty, List Names

#### NOTE:

Data is not available until an order is received by your fulfillment house and has activated the Keys; you need to be aware of this timing if you want to assign keys. When keys are not active, you can set up the rest of the campaign so that when they do become active you can assign them and begin using IMT Reporting immediately.

#### Develop a Strategy

It is recommended that you map out how you will organize, define and name your sources, campaigns and panels before using Admin. Organizing this before hand will make for consistent reporting.

While adding records, you will be required to input certain fields (i.e. **Campaign Name**). Other fields may be optional (i.e. **Campaign Type**). The definition of a campaign may also vary depending on the source of business or the reporting requirements. Your choice of optional fields to populate may depend on the source of business. Generally, the more fields you populate, the richer your reporting capabilities.

#### Familiarize yourself with IMT Reporting

Before getting started, you might want to practice navigating the system. This will help you to better understand the database, and make administration of your campaigns easier and more intuitive. It is helpful to understand how IMT organizes information into hierarchical groups as follows:

1. **GROUP** (i.e. Company Division)
2. **PRODUCT** (i.e. Magazine Title or Product Name)
3. **SOURCE CATEGORY** (i.e. DTP, etc.)
4. **SOURCE** (i.e. Inserts, etc.)
5. **CAMPAIGN NAME** (i.e. 2000-01)
6. **PANEL NAME** (i.e. 01. Sub 12/\$12)
7. **PROMOTION KEY** (i.e. 41B2\_12/01/2000)

#### One-on-One Walk-Through of IMT Reporting

If you want to see examples of campaigns that are already setup in the system, you might want to look at IMT's demo database. For a one-on-one walk-through of IMT's demo database, please request a demo username and password at: [ClientServices@imtnetwork.com](mailto:ClientServices@imtnetwork.com), name the subject of your email: **demo request** or call us at 415.352.2370.

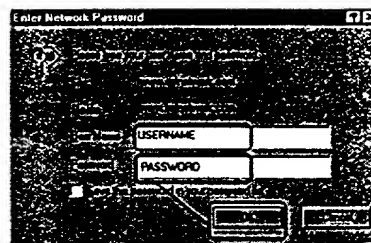
**Step – 1:** Open your **Browser** and input IMT's URL (Internet Home Page address) (**NOTE:** once you are viewing the home page, right click your mouse and select **Create Shortcut**):



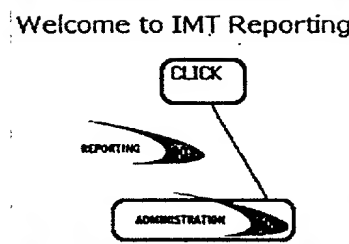
**Step – 2:** Click **Client Center** to reach secure log in box:



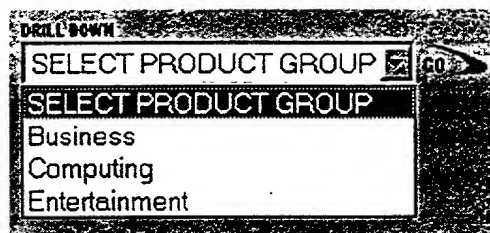
**Step – 3:** Input your **Username & Password**: (**NOTE:** case sensitive; i.e. a or A)



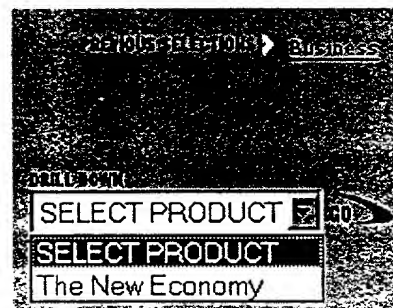
**Step – 4:** Click on **Administration** to begin building and managing campaigns:

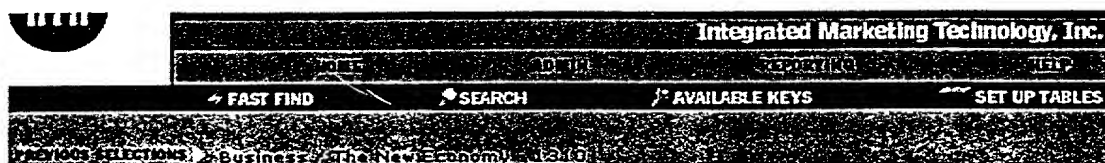


**Step – 5:** **Select Product Group** (i.e. Business):



**Step – 6:** **Select Product** (i.e. The New Economy):

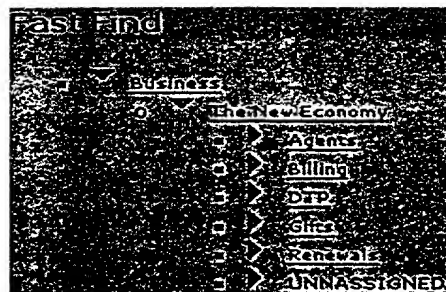




Once you have accessed IMT Reporting, you will now be able to view the Tool Bar (located across the top of the screen). Before building campaigns, you should familiarize yourself with the Tool Bar features and understand how they will assist with the administration and reporting processes.

**Fast Find** – Allows you to navigate the database using a drill-down tree.

- **Drill-down** into Product Group, Product, Source Category, Source, Campaign, and Panel by clicking on ► to open the next table below.
- **Open view** of table by clicking table name (i.e. **DTP** to open tables in DTP)



**Search** – Allows you to search for specific sources, campaigns and panels by group name and key words.

- Select one **Title / Product**
  - Select **Source Category**
  - Select **Source**
- NOTE:** the % will open the search parameters
- Select **Campaign** – input before %
  - Select **Panel** – input before %
  - Select Break on **Campaign** – list of campaigns
- or
- Select Break on **Panel** – list of panels within a campaign

<b>TITLE</b>	The New Economy
<b>SOURCE CATEGORY</b>	DTP
<b>SOURCE</b>	Insert Cards
<b>CAMPAIGN</b>	Put your search here%
<b>PANEL</b>	Panel Name to search%

Break on

☒ CAMPAIGN

☐ SUBMIT

**Available Keys** – Allows you to filter and search the available key list (List of all "un-assigned" keys for the product you are accessing).

- Start with position #1 – **Source Code** (i.e. 4 = Insert Cards)
- Click **GO**

**NOTE:** Each position (1-10) will accept an alphanumeric entry; use all 10 or a minimum of 1 to filter your search.

Filter Keys By Position

1 2 3 4 5 6 7 8 9 10

4

Product: The New Economy  
1010

Change Product

☒ FILTER ALL ☐ CLEAR ALL

AVAILABLE KEYS	
<input type="checkbox"/>	4005_04/24/2000*
<input type="checkbox"/>	4005_6_04/24/2000*
<input type="checkbox"/>	4182_08/21/2000*

Chart of **Setup Tables** and associated database tables, to which they apply:

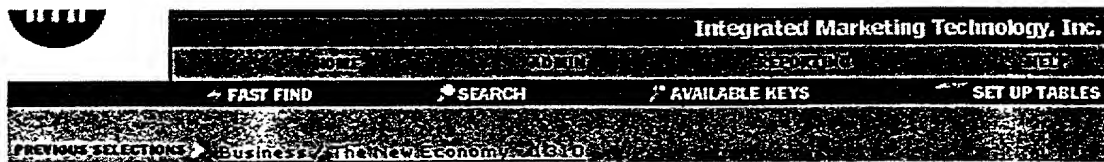
- **Create** the tables shaded below prior to building campaigns
- **C = Corporate Setup Table:** will assign across a group of products
- **P = Product Setup Table:** will assign across individual products

In addition to allowing you to associate keys to sources, campaign and panels, IMT Reporting provides the ability to attach additional information to a source key/promotion key (i.e. user may want to associate specific list selections to direct mail source keys).

SETUP TABLE	DEFINITION	EXAMPLE	CAMPAIGN TABLE	PANEL TABLE	KEY TABLE	C/P
<b>AutoAdmin Parameters</b>	<i>See the pages listed</i>	<ul style="list-style-type: none"> <li>• Pages: 19 - 28</li> </ul>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	P
<b>Campaign Status:</b>	Define the activity of a Campaign	<ul style="list-style-type: none"> <li>• Active</li> <li>• Closed</li> <li>• Pending</li> </ul>	<input checked="" type="checkbox"/>			P
<b>Campaign Type:</b>	A broad type classification of a Campaign which is user defined and can differ between sources	<ul style="list-style-type: none"> <li>• Winter DM</li> <li>• Hotline</li> </ul>	<input checked="" type="checkbox"/>			P
<b>Expense Set:</b>	A set of expenses which can be applied to a Campaign, Panel, or Key	<ul style="list-style-type: none"> <li>• 2000-01 DM 6x9 Package Exp.</li> <li>• Triple Insert Exp.</li> </ul>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	P
<b>List Category:</b>	A broad category of related lists	<ul style="list-style-type: none"> <li>• Home</li> <li>• Sports</li> </ul>			<input checked="" type="checkbox"/>	P
<b>List Name:</b>	ID to define the list	<ul style="list-style-type: none"> <li>• Tattooing Weekly</li> <li>• Money Monthly</li> </ul>			<input checked="" type="checkbox"/>	P
<b>List Segment:</b>	A specific selection from a list	<ul style="list-style-type: none"> <li>• Paid Subs</li> <li>• 3 Month Buyers</li> </ul>			<input checked="" type="checkbox"/>	P
<b>Panel Package:</b>	Define the type/style of a package	<ul style="list-style-type: none"> <li>• 6x9 Control</li> <li>• #10 Sweeps</li> </ul>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	P
<b>Panel Subtype:</b>	Further define another field for a Panel	<ul style="list-style-type: none"> <li>• Sweepstakes</li> <li>• Newsstand</li> </ul>		<input checked="" type="checkbox"/>		C
<b>Panel Test Type:</b>	Define the type of test	<ul style="list-style-type: none"> <li>• Price</li> <li>• Offer</li> </ul>		<input checked="" type="checkbox"/>		C
<b>Panel Type:</b>	Define the type of Panel	<ul style="list-style-type: none"> <li>• Control</li> <li>• Test</li> </ul>		<input checked="" type="checkbox"/>		C
<b>Revenue Set:</b>	Revenue over and above subscription revenue, applied to a Campaign, Panel, or Key	<ul style="list-style-type: none"> <li>• List Rental Revenue</li> <li>• Advertising Revenue</li> </ul>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	P
<b>Source Description:</b>	Define a specific channel of sale	<ul style="list-style-type: none"> <li>• Direct Mail</li> <li>• Renewals</li> </ul>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	C
<b>Source ID Description:</b>	Allow you to define a source beyond the one character fulfillment source description. Descriptions are used in reporting.	<ul style="list-style-type: none"> <li>• D or 2 = Direct Mail</li> </ul>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	C/P

**NOTE:**

To define one of the **Setup Table** fields – review instructions on page 10.

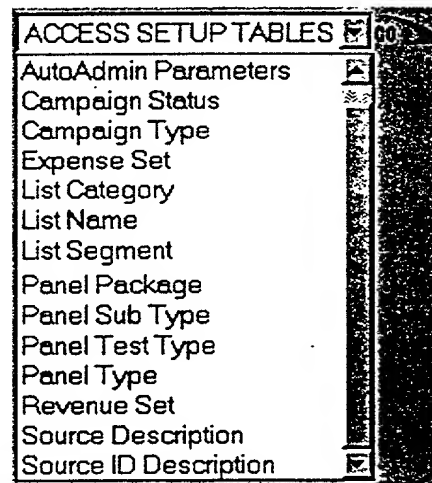


Each product has unique features and the **Setup Tables** are used to define campaigns, panels, or keys with detailed information that will allow your reports to produce a more valuable result (stronger and easier to read).

#### Step – 1 Access Setup Tables

- Click **Setup Tables** located on the **IMT Tool Bar** (NOTE: You can access setup tables at any point in the administration process, after you have selected a product; as shown in tool bar figure at the top of this page).

**Step – 2** Select the title / product and setup table you wish to edit. When product is not selected, a prompt will appear.



**Step – 3** **ADD** new record(s) by clicking **ADD** button (located at the top of each Table).

- Input your definition into the available field.
- Submit** and this will populate that Table so you can make a selection from the new record(s) when you build your campaign(s).

ADD	
Campaign Type	Description
2000 Fall	
2000 Q1	
2000 Q2	

#### NOTE:

- Click on **Change Product ID**, this link will allow you to move from product to product while still being in the selected Setup Table.
- Click on **Return to Setup Table Selection Screen**, this link will take you back to Step-2



**Step – 4** **EDIT** existing records by clicking **EDIT** button (located at the left of each Table definition that exists)

- Input changes
- Submit**

The screenshot shows an 'EDIT' form for 'CAMPAIGN TYPE NAME'. It has a 'SUBMIT' button and a 'DELETE' button. Below the buttons is a 'Return to Campaign Type Setup Table' link. The form contains two input fields: 'CAMPAIGN TYPE NAME' with the value '2000 Q1' and 'DESCRIPTION' which is empty.

**1. Select a Product Group, Product & Source Category (i.e. Business/The New Economy/DTP)**

**2. Define Source(s)**

- Locate the **IMT Tool Bar** and click on **Setup Table** (window #2 will open)
- Select **Source Description**
- **Add** or **Edit** per instructions on the previous page 10
- **Submit** to enter and save your information
- Close Setup Table (#2) window

**3. Add new Source(s) (return to main window #1)**

- Click **Add New**
- Select **Source Name**
- Input a **Source Indicator** (the one digit alpha-numeric code that corresponds to the first position of the source key provided by the fulfillment house (i.e. 4 = Inserts or 0 = Direct Mail))
- **Submit** to enter and save your information

**TIPS:**

- **SUBMIT** button - apply changes to the database.
- **ADD** button - add a record to the database
- **EDIT** button - edit a record in the database.
- **RETURN / BACK** to a previous screen/page use (Click on the underlined link and return)
- **NOTE: DELETING A SOURCE;** you must contact IMT directly via [support@imtnetwork.com](mailto:support@imtnetwork.com)



## STEP 2 – ADD OR COPY CAMPAIGNS

1. Select a **Source** (i.e. Inserts, etc.)

PREVIOUS SELECTIONS: Business / The New Economy / 310 / DTP

DRILL DOWN

SELECT SOURCE ☒ TO

SELECT SOURCE

Direct Mail

Insert Cards

Internet

SOURCE NAME	SOURCE INDICATOR
Direct Mail	0
Insert Cards	4

## 2. Add new Campaign(s) (Fig 3.10)

- Click **Add**
- Input **Campaign Name** (i.e. 2000-12) (NOTE: use year before month to better sort multiple campaigns)
- Input **Start Date** (i.e. 12/1/00)
- Input optional data: **Campaign Type, Status, Expense Set, Revenue Set, Budgeted Gross Pct, Budgeted Volume**
- Submit** to enter and save your information

ADD

Return to Campaign

CAMPAIGN NAME	CAMPAIGN START DATE
2000 - 01	1/1/00

## 3. Copy feature:

- Select the campaigns to **Copy** (you can choose more than one at a time to copy) with ☒ at the left of the campaign name
- Select **Copy** button to duplicate campaign(s) and panel(s) associated with campaigns copied (your copies will appear at the top of the campaign table (i.e. Copy 1 of 2000-12). **Edit** the copied **campaign name** (i.e. 2001-12)(NOTE: Campaign names are unique to each source name).
- Edit** Once you ☒ the record(s) you wish to edit, click the Edit button to go to the Edit screen and make changes.
- Submit** to enter and save your information

Show Group Edit

Check/Uncheck	Add	Edit	Copy	Delete
CAMPAIGN NAME	CAMPAIGN DATE	CAMPAIGN TYPE	CAMPAIGN	
<input checked="" type="checkbox"/> 2000-05	5/1/2000	Undefined	Undefined	
<input type="checkbox"/> 2000-04	4/1/2000	Undefined	Undefined	
<input type="checkbox"/> 2000-03	3/1/2000	Undefined	Undefined	

## TIPS:

- EDIT** button – edit records in the database
- COPY** button - copy the campaigns & panels (not keys).
- DELETE** button - delete a campaign (and its associated panels and keys) or panels and keys separately.
- RETURN / BACK** to a previous screen/page use (Click on the underlined link and return)

PREVIOUS SELECTIONS: Business / The New Economy / 310 / DTP / Insert Cards



## 1. Select a Campaign (i.e. 2000-01)

DRILL DOWN

SELECT CAMPAIGN

SELECT CAMPAIGN

2000-05  
2000-04  
2000-03  
2000-02  
2000-01

CAMPAIGN NAME	CAMPAIGN
<input type="checkbox"/> 2000-05	5/1/2000

## 2. Add new Panel(s)

- Click
- Input **Panel** name (i.e. Control, Price test, etc.)
- Input Optional Data: **Panel Start Date, Panel Type, SubType, Test Type, Package, Expense Set, Revenue Set**
- Submit** to enter and save your information

ADD

PANEL NAME	PANEL START DATE
<input type="text"/>	<input type="text"/>

## 3. Copy feature:

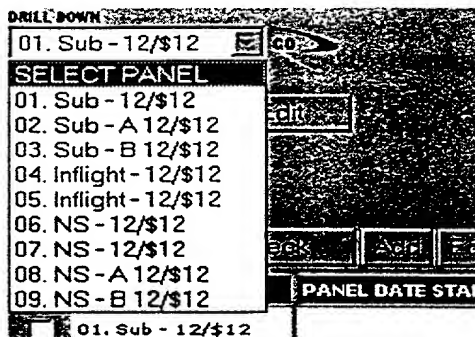
- Select the panels to **Copy** (you can choose more than one at a time to copy) with ☒ at the left of the panel name
- Select  button to duplicate panel(s) (copies will appear at the bottom of the panel table (i.e. Copy 1 of 01. Sub). **Edit** the copied **panel name** (NOTE: Panel names are unique to each source name).
- Once you ☒ the record(s) you wish to edit, click the Edit button to go to the Edit screen and make changes.
- Submit** to enter and save your information

<input type="button" value="Check/Uncheck"/>	<input type="button" value="Add"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	
	PANEL NAME	PANEL DATE START	PANEL TYPE		
<input checked="" type="checkbox"/>	01. Sub - 12/\$12		Subscriber	B	
<input type="checkbox"/>	02. Sub - A 12/\$12		Subscriber	B	
<input type="checkbox"/>	03. Sub - B 12/\$12		Subscriber	B	
<input type="checkbox"/>	04. Inflight - 12/\$12		Inflight	B	
<input type="checkbox"/>	05. Inflight - 12/\$12		Inflight	B	
<input type="checkbox"/>	06. NS - 12/\$12		Newsstand	B	
<input type="checkbox"/>	07. NS - 12/\$12		Newsstand	B	
<input type="checkbox"/>	08. NS - A 12/\$12		Newsstand	B	
<input type="checkbox"/>	09. NS - B 12/\$12		Newsstand	B	
<input type="checkbox"/>	Copy 1 of 01. Sub - 12/\$12		Subscriber	B	

## TIPS:

- EDIT** button - edit records in the database
- COPY** button - copy the campaign & panels (not keys).
- DELETE** button - delete a campaign and its associated panels and keys.
- RETURN / BACK** to a previous screen/page use (Click on the underlined link and return)

PREVIOUS SELECTIONS Business / The New Economy - 10/10/01 / ODP / Insert Cards / 2000-01



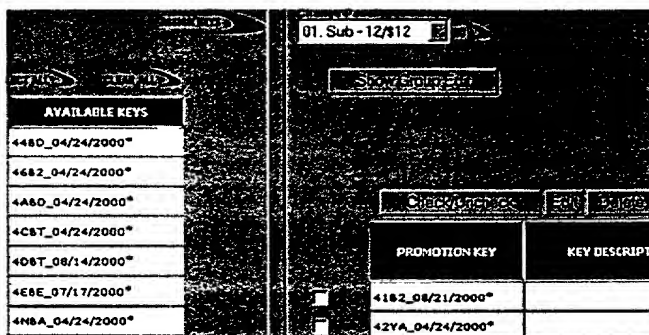
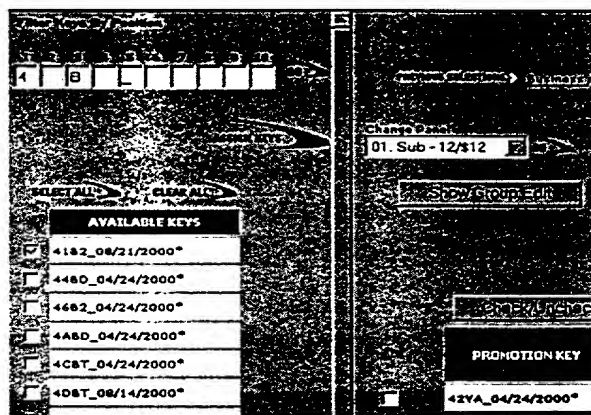
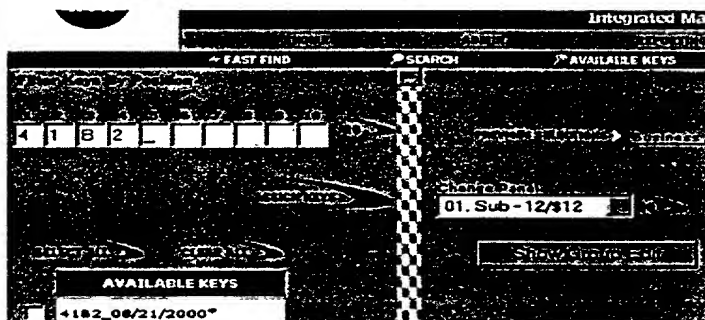
1. Select a **Panel** (i.e. 01.Sub – 12/\$12)

2. Assign new **Key(s)** - Confirm the correct **Panel** in view to assign key(s)

### 3. Filter Keys By Position

- Input one or more of the **10 key** position fields (i.e. 41B2\_08/21/2000), use less fields to view more keys in the available keys table)
- Click **Go** to view the specific list of available keys within given range
- Select the **Keys** you are assigning to the **Panel**
- Select ☒ at the left of each individual key
- Click **Select All** to select all available keys in view
- Click **Clear All** to de-select all available keys in view
- Click **Assign Keys** when your selections are complete (your selected keys will now be assigned / viewed in the **Key Table** in the selected **Panel**)

4. When assigning Keys to more than one **Panel** within a campaign, select **Change Panel** and repeat steps above



#### NOTE:

For more information on assigning **Insert Card Keys** and using the **Draw/Sale/Subtotal Flag** fields, go to **Page 29**



The **Group Edit** and **Multi-Edit** features in Admin allow the user to more easily edit campaigns, panels, or keys. For example: Make a universal change on all panels, use the **Group Edit** feature. Or, if there are several independent fields to change on each panel, you could use the **Multi-Edit** feature.

**GROUP EDIT:**

**Step - 1** Click the **Show Group Edit** button. This will bring up the group edit bar with the same headings as the columns in the campaign, panel or keys tables. To close the Group Edit screen, click on **Hide Group Edit**.

**Step - 2** Check the ☒ next to the campaigns, panels, or keys you wish to edit. Or you can use **Check/Uncheck** - This button will select each row of data by placing a check mark (☒) in all the white boxes. Click it again and it will clear the selection. You can also use this button to toggle selected ☒/☐.

**Step - 3** In the Group Edit section at the top, make the change to the field(s) you wish to edit.

Group Edit				
IGN DATE	CAMPAIGN TYPE	CAMPAIGN STATUS	EXPENSE SET	REVENUE
e)	(No Change) <input checked="" type="checkbox"/>	(No Change) <input checked="" type="checkbox"/>	(No Change) <input checked="" type="checkbox"/>	(No Change) <input checked="" type="checkbox"/>

**Step - 4** Once you've made the change, click **Submit Changes**. You will see that the edits you requested are now reflected below in the records you chose to edit.

**MULTI-EDIT:**

**Step - 1** Check the ☒ next to any row(s) you wish to edit. Or you can use the **Check/Uncheck** button described above.

**Step - 2** Select **Edit**

Page 1 - Records 1-5 of 5

[Return to Menu](#)

				Multi Edit	
CAMPAIGN NAME	CAMPAIGN START DATE	CAMPAIGN TYPE	CAMPAIGN STATUS		EX
2000-05	5/1/2000	2000 Fall	<input checked="" type="checkbox"/> Active <input checked="" type="checkbox"/>	Undefined	
2000-04	4/1/2000	2000 Fall	<input checked="" type="checkbox"/> Active <input checked="" type="checkbox"/>	Undefined	
2000-03	3/1/2000	2000 Fall	<input checked="" type="checkbox"/> Active <input checked="" type="checkbox"/>	Undefined	
2000-02	2/1/2000	2000 Fall	<input checked="" type="checkbox"/> Active <input checked="" type="checkbox"/>	Undefined	
2000-01	1/1/2000	2000 Fall	<input checked="" type="checkbox"/> Active <input checked="" type="checkbox"/>	Undefined	


**Submit Changes**

**NOTE:**

The system allows you to select and edit up to 20 records at a time, however the actual **Edit** screen will only show five of your selections at one time. Make the necessary changes, click **Submit Changes** button, and this will save changes and take you to the next screen to edit remaining selections. When all selections have been edited, the system will automatically take you back to the initial screen.

## 1. Select a Source (i.e. Inserts)



## 2. Edit one Campaign

- Select ☒ existing campaign to edit, click 
- **Edit Campaign Name** (i.e. 2000-01) (**NOTE:** use year before month to better sort multiple campaigns)
- **Edit Start Date** (i.e. 12/1/00)
- **Edit optional data: Campaign Type, Status, Expense Set, Revenue Set, Budgeted Gross Pct, Budgeted Volume**
- **Submit** to enter and save your information

## 3. Group Edit or Multi-Edit existing Campaign

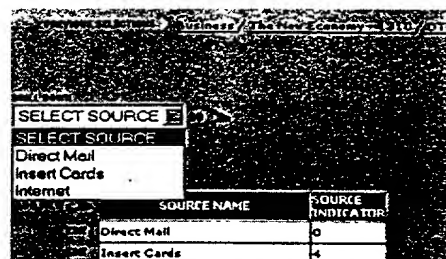
## Option A: Click



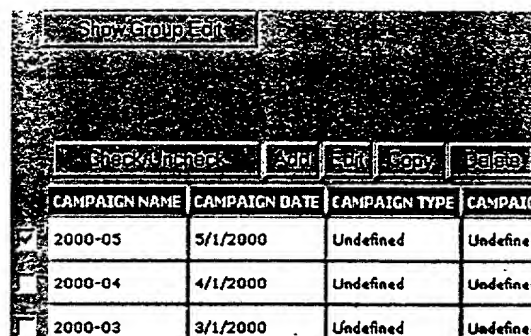
- Click  button to auto select/unselect (☒) campaigns to edit
- or
- Select/unselect individually by clicking ☒ at left of campaign name, confirm the campaigns to edit (as a group) with ☒ at the left of the campaign name. Make your edit(s) in the open fields (Group Edit Bar above the Campaign Table).  to enter and save your information.

## Option B: Multi-Edit feature:

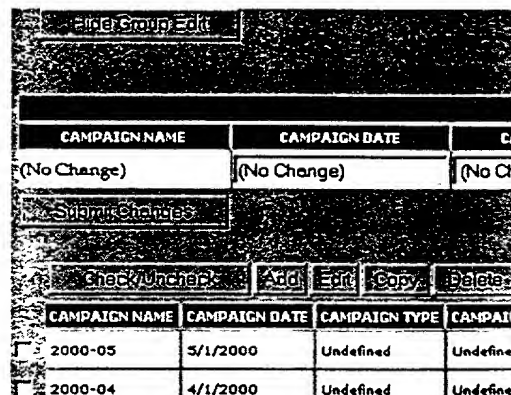
- Select the campaigns to edit with ☒ at the left of the campaign name
- Click **Edit** button above campaign table to view campaign(s) to edit
- Make your edit(s) in the open fields associated with the campaign(s) to edit (**NOTE:** only 5 Campaigns are in view at one time, while using **Multi-Edit**)
- Click **Submit Changes** button to save changes and take you to the next screen to edit remaining selections. When all selections have been edited, the system will automatically take you back to the initial screen.



SOURCE NAME	SOURCE INDICATOR
Direct Mail	0
Insert Cards	1



CAMPAIGN NAME	CAMPAIGN DATE	CAMPAIGN TYPE	CAMPAIGN
2000-05	5/1/2000	Undefined	Undefined
2000-04	4/1/2000	Undefined	Undefined
2000-03	3/1/2000	Undefined	Undefined



CAMPAIGN NAME	CAMPAIGN DATE	CAMPAIGN
(No Change)	(No Change)	(No Ch
2000-05	5/1/2000	Undefined
2000-04	4/1/2000	Undefined

## TIPS:

- **EDIT** button – edit records in the database
- **COPY** button – copy the campaign & panels (not keys).
- **DELETE** button – delete a campaign and its associated panels and keys.
- **RETURN / BACK** to a previous screen/page use (Click on the underlined link and return)

PREVIOUS SELECTIONS Business / The New Economy / 310 / DTP / Insert Cards

1. Select a Campaign (i.e. 2000-12)

2. Edit Panel

- Select ☒ existing Panel to edit, click **Edit**
- Edit Panel name (i.e. Control, Price test, etc.)
- Edit Optional Data: Panel Start Date, Panel Type, SubType, Test Type, Package, Expense Set, and Revenue Set
- Submit to enter and save your information

3. Group Edit or Multi-Edit existing Panel

Option A: Group Edit feature:

- Click **Show Group Edit** button below Select Panel field
- Select **Check/Uncheck** button to auto select/un-select (☒) panels to edit

or

- Select/unselect individually by clicking ☒ at left of panel name
- Confirm the panels to edit (as a group) with ☒ at the left of the panel name
- Make your edit(s) in the open fields (Group Edit Bar above the Panel Table)
- Click **Submit Changes** to enter and save your information

PANEL NAME	PANEL DATE START	PANEL TYPE
01. Sub - 12/\$12		Subscriber
02. Sub - A 12/\$12		Subscriber
03. Sub - B 12/\$12		Subscriber
04. Inflight - 12/\$12		Inflight
05. Inflight - 12/\$12		Inflight
06. NS - 12/\$12		Newstand
07. NS - 12/\$12		Newstand
08. NS - A 12/\$12		Newstand
09. NS - B 12/\$12		Newstand
Copy 1 of 01. Sub - 12/\$12		Subscriber

Option B: Multi-Edit feature:

- Select the panels to edit with ☒ at the left of the panel name
- Click **Edit** button above panel table to view panel(s) to edit
- Make your edit(s) in the open fields associated with the panel(s) to edit (**NOTE: only 5 Panels are in view at one time, while using Multi-Edit**)
- Click **Submit Changes** button to save changes and take you to the next screen to edit remaining selections. When all selections have been edited, the system will automatically take you back to the initial screen.

PANEL NAME	PANEL DATE START
<input checked="" type="checkbox"/> 01. Sub - 12/\$12	
<input checked="" type="checkbox"/> 02. Sub - A 12/\$12	
<input type="checkbox"/> 03. Sub - B 12/\$12	
<input checked="" type="checkbox"/> 04. Inflight - 12/\$12	
<input checked="" type="checkbox"/> 05. Inflight - 12/\$12	

#### TIPS:

- **EDIT** button – edit records in the database
- **COPY** button - copy the campaign & panels (not keys).
- **DELETE** button - delete a campaign and its associated panels and keys.
- **RETURN / BACK** to a previous screen/page use (Click on the underlined link and return)

[PREVIOUS SELECTIONS - Business / The New Economy - B10 / DTP / Insert Cards / 2000-01](#)

1. Select a Panel (i.e. control)

2. Edit one Key

- Select ☒ existing Key to edit, click **Edit**
- Edit Key name (i.e. Control, Price test, etc.)
- Edit Optional Data: **Panel Start Date, Panel Type, SubType, Test Type, Package, Expense Set, and Revenue Set**
- Submit to enter and save your information

3. Group Edit or Multi-Edit assigned Key

Option – A: Group Edit feature

- Click **Show/Group Edit** button below the Change Panel/Selected Panel field
  - Select **Check/Uncheck** button to auto select/un-select (☒) Keys to edit
- or
- Select/unselect individually by clicking ☒ at left of key name/code
  - Confirm the keys to edit (as a group) with ☒ at the left of the key name
  - Make your edit(s) in the open fields (Group Edit Bar above the Key Table)
  - Click **Submit Changes** to enter and save your information

Option – B: Multi-Edit feature:

- Select the keys to edit (as a group or individually) with ☒ at the left of the key name
- Click **Edit** button above key table to view key(s) to edit
- Make your edit(s) in the open fields associated with the key(s) to edit (**NOTE: only 5 Panels are in view at one time, while performing Multi-Edit**)
- Click **Submit Changes** button to save changes and take you to the next screen to edit remaining selections. When all selections have been edited, the system will automatically take you back to the initial screen.

DRILL DOWN

01. Sub - 12/\$12 ☒ GO

**SELECT PANEL**

01. Sub - 12/\$12  
02. Sub - A 12/\$12  
03. Sub - B 12/\$12  
04. Inflight - 12/\$12  
05. Inflight - 12/\$12  
06. NS - 12/\$12  
07. NS - 12/\$12  
08. NS - A 12/\$12  
09. NS - B 12/\$12

Back Add Edit

PANEL DATE START

☒ 01. Sub - 12/\$12

Change Panel: 01. Sub - 12/\$12 ☒ GO

**Show/Group Edit**

**Check/Uncheck** **Edit** **Data**

PROMOTION KEY	KEY DESCRIPTION
<input checked="" type="checkbox"/> 4182_08/21/2000*	
<input type="checkbox"/> 42YA_04/24/2000*	

Change Panel: 01. Sub - 12/\$12 ☒ GO

**Hide Group Edit**

KEY DESCRIPTION	MAIL QTY OVERRIDE / DRA
(No Change)	(No Change)

**Submit Changes**

**Check/Uncheck** **Edit** **Data**

PROMOTION KEY	KEY DESCRIPTION
<input checked="" type="checkbox"/> 4182_08/21/2000*	

**TIPS:**

- **EDIT** button – edit records in the database
- **COPY** button - copy the campaign & panels (not keys).
- **DELETE** button - delete a campaign and its associated panels and keys.
- **RETURN / BACK** to a previous screen/page use (Click on the underlined link and return)

PREVIOUS SELECTIONS: Business / The New Economy / 1310 / DTP / Insert Cards / 2000-01 / 01:Sub=12/\$12

**IMT AutoAdmin** has been created to minimize the amount of time spent on setting up campaigns in IMT Reporting. The idea behind AutoAdmin is that certain keys have a solid structure and therefore can be assigned based on the design of the key. Bills and Renewals are good examples of sources with a solid key structure. Fulfillment houses generally maintain uniformity across these two sources.

IMT is able to look to the appropriate position in each key to pull relevant information from the key, and then designate an appropriate source, campaign and panel. AutoAdmin within IMT Reporting refers to the process whereby source keys are automatically assigned to Sources, Campaigns and Panels without manual administration (see Building Campaigns section). This is accomplished through a series of user-defined setup tables that serve as instructions for interpreting source keys. These tables serve as filters that provide the system with basic instruction.

### Does AutoAdmin replace manual administration?

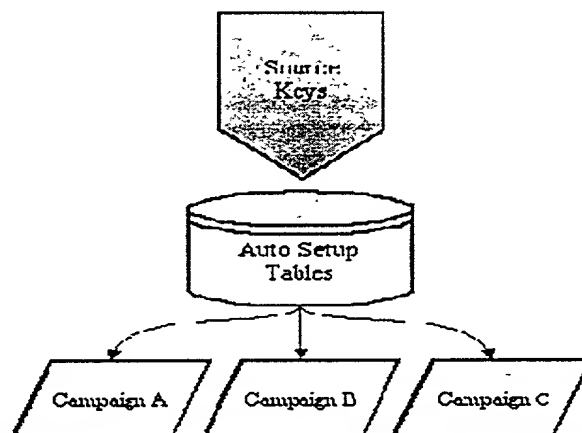
No, AutoAdmin Setup is an enhancement to IMT Reporting. If you prefer to setup your campaigns manually, then you can continue to do so. In fact there are some sources where manually administration is preferable or required. NOTE: Bills & Renewals generally have the same panel name (i.e. control) across multiple campaigns.

### Does Auto Setup work for any source?

Yes, AutoAdmin can be used for any source of business. However, in order for AutoAdmin to work most effectively a clearly defined source key logic must exist. As a minimum requirement, specific positions from a source key must distinguish the source and campaign. Billing and Renewals source, which often account for the highest number of source keys, typically utilize a standardized key logic that is well suited to AutoAdmin. For other sources of business that are not as standardized, manual administration of campaigns may be preferable or required.

### How do I get started? (Where is it located?)

To get started, you will need to learn more about how the AutoAdmin Setup tables work and how to populate these tables correctly. To access **AutoAdmin** click on the **Setup Tables** (located on the IMT Tool Bar), then select **AutoAdmin** from the drop down menu.



#### NOTE:

Once these tables have been populated, the system will automatically assign keys to the correct source, campaign and panel as they come on the file. It is important, however, to check that keys are assigned correctly in order to ensure accuracy in reporting. There are checking reports available, which will aid in this process. For Bills, use the "Billing Source Key Assignment - Checking Report"; for Renewals, use the "Renewals Source Key Assignment - Checking Report".



## HOW TABLES WORK

There are four Auto Setup tables, which together control how source keys are associated to sources, campaigns and panels.

TABLE	PURPOSE	REQUIRED?
Auto Setup Schemes	The Auto Setup Schemes table allows a user to create filters through which source keys are passed. The primary component of a scheme is called the <i>setup mask</i> . A setup mask identifies which position(s) of a source key represent the source, campaign, panel or prior source.	YES
Campaign Name Definitions	This Campaign Name Definition table provides a user the opportunity to convert campaign codes to names that are more descriptive. Campaign codes are based upon the source key positions that represent the campaign.	NO
Renewal Prior Source Definitions	This table is only used for renewal sources, and only in cases when a source is split. For example, if all conversions source keys begin with a source indicator "6", you may wish to split out DTP and agent conversions separately based upon prior source.	NO
Panel Name Definitions	This table is only used in conjunction with setup schemes that identify panel splits. If a panel split is identified in a setup scheme, then panel codes can be given more descriptive names by adding records to this table, and then applying the panel-naming scheme to a campaign in the campaign names definitions table.	NO

## NOTE:

Not all the Auto Setup tables need to be populated for the system to automatically assign keys to sources and campaigns. In fact, only the Auto Setup Schemes table is required for Auto Setup to work; the other tables provide a means of controlling more precisely how campaigns and panels are named, or in the case of renewals provide the ability to split a source.

AutoSetup Schemes						
ADD NEW	Source	Source Indicator	Select Date From	Select Date To	Setup Mask	Expense Set
EDIT	Billing	B	1/1/1998	12/31/2005	SCCHCCHHHHHHHHP	Undefined

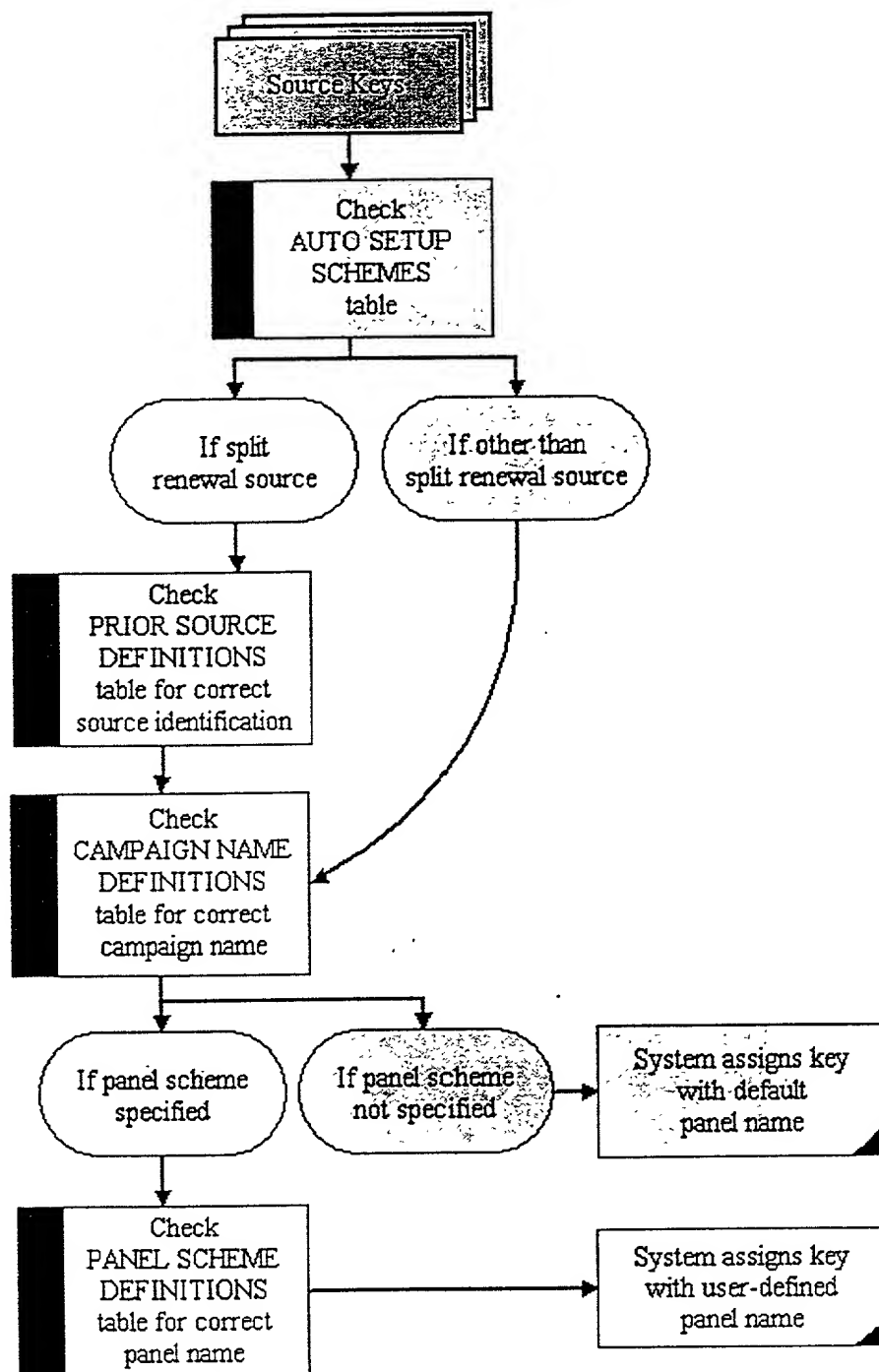
Campaign Name Definitions				
ADD NEW	Source Category	Campaign Code	Campaign Description	Panel Scheme
EDIT	Billing	AX00	00 Credit Period AX00	8/1/2000

Renewal Prior Source Definitions					
ADD NEW	Source	Source Indicator	Prior Source Indicator	Valid From	Valid To
EDIT	Agent Conversions	6	A,B,C,D,E,F,G,H,J,K,L,M,N,P	1/1/1999	12/31/2005

Panel Name Definitions		
ADD NEW	Panel Scheme	Panel Codes
EDIT	Ack Retest	P,Z,U,D,G,J,S,T,Q,C,B,Z,1,2,3,9,*
		Control



The schematic below shows the sequence of checks that Auto Setup makes in order to correctly associate source keys to sources, campaigns and panels.





IMT Reporting will divide source keys into four broad categories (Source Category):

- DTP (direct to publisher)
- Renewal
- Gift
- Billing

Depending on the fulfillment company, the source category, and IMT's summary procedures, source keys may have different formats. Below is a chart to help identify and interpret source keys based on fulfillment company.

EDS / CENTROBE		
TYPE	EXAMPLE	DESCRIPTION
DTP	0SUA_20000124	Position 1 = Source Positions 2-4 = Open Positions 6-13 = Date (yyyymmdd)  NOTE: An asterisk (*) appended to a key indicates that the key was received without a select date specified. As a result, IMT appends the process date to the key for id purposes.
RENEWAL	7AK1_6_526	Position 1 = Source Position 2 = Issue or Package Position 3 = Issue or Package Position 4 = Effort Position 6 = Prior Source Position 7-9 = Expire
GIFT	2CL5_G_20001215	Position 1 = Source Position 2-4 = Open Position 6 = Gift Identifier Position 8-15 = Date (yyyymmdd)  NOTE: An asterisk (*) appended to a key indicates that the key was received without a select date specified. As a result, IMT appends the process date to the key for id purposes.
BILLING	B52_990321_4_1A	Position 1 = Source Position 2-3 = Bill key Position 5-10 = Date (yyymmdd) Position 12 = Current Source Position 14 = Effort Position 15 = Package



## SOURCE KEY ANATOMY - CONTINUED

PALM COAST DATA		
TYPE	EXAMPLE	DESCRIPTION
DTP	28M25A_1999	Position 1 = Source Position 2-6 = Open Position 8-11 = Date (yyyy)
RENEWAL	3E9H01_1999	Position 1 = Source Position 2 = Prior Source Position 3-4 = Issue Position 5 = Package Position 6 = Effort
GIFT	D9NC11_1999	Position 1 = Source Position 2-6 = Open Position 8-11 = Date (yyyy)
BILLING	B2_930_00	Position 1 = Source Position 2 = Prior Source Position 4 = Year (y) Position 5-6 = Credit Period Position 8-9 = Effort

CDS		
TYPE	EXAMPLE	DESCRIPTION
DTP	DP2F001_010199	Position 1 = Source Position 2-4 = Open Position 5-7 = Split Position 9-14 = Date (mmddyy)
RENEWAL	RU21000_010199	Position 1 = Source Position 2 = Issue Position 3 = Prior Source Position 4 = Effort Position 5-7 = Split Position 9-14 = Date (mmddyy)
GIFT	YTA1000_010199	Position 1 = Source Position 2-4 = Open Position 5-7 = Split Position 9-14 = Date (mmddyy)
BILLING	BP21000_010199	Position 1 = Source Position 2 = Credit Period Position 3 = Current Source Position 4 = Effort Position 5-7 = Split Position 9-14 = Date (mmddyy)

**AutoAdmin Setup Schemes** table tells the system what sources to look for to automatically setup campaigns. You define the criteria to identify and assign source keys to a campaign. The system will work even if this is the only table you setup. (However, to do proper naming of campaigns and splitting of panels, you should use all the tables. For example, if you only filled out this table, then your campaigns and panels would be named based on the fulfillment codes: Campaign = Expire 521, Panel = Panel A. If you filled out the Campaign Name Definition table and Panel Schemes table, then you could have the system name them: Campaign = 2000-01, Panel = Control).

The **Setup Mask** is the most important element of any Setup Scheme. The Setup Mask is simply a string of alpha characters used to identify the parts of a source key. IMT has created a simple set of characters to identify source, campaign, panel/split, and prior source (renewals only). By stringing these characters together to form a Setup Mask, you provide our system with all the information required to automatically identify and create campaigns.

Setup Mask Legend		
Identifier	Description	Examples
S	Source	B = Billing 1 = Renewals
C	Campaign, Expire or Credit Period	125 = Expire 125 (Jan-2000) 001 = Jan 2000 Expire
P	Package/Split	0 = Renewal Control 1 = Renewal Test
Z	Prior Source	0 = Direct Mail 6 = Conversions A = AFP
x	Placeholder	Used to mark a position on a source key that does NOT identify Source, Campaign, Panel/Split, or Prior/Current Source

**NOTE:**

Understanding how your source keys are constructed is a pre-requisite to creating Auto Setup Schemes. Since IMT will often enhance source keys (in order to provide better reporting capabilities), you should familiarize yourself with IMT's source key structure.

AutoSetup Schemes						
	Source	Source Indicator	Select Date From	Select Date To	Setup Mask	Expense Set
EDIT	Billing	B	1/1/1998	12/31/2005	SCCnCCxxxxxxxxxP	Undefined
EDIT	Billing	B	1/1/1998	12/31/2005	SCCnCCxxxxxxxxxP	Undefined
EDIT	Renewals 1x	1	1/1/1998	12/31/2005	SxPnxxZnCCC	Undefined
EDIT	Renewals 2+	1	1/1/1998	12/31/2005	SxPnxxZnCCC	Undefined
EDIT	Agent Conversions	6	1/1/1998	12/31/2005	SxPnxxZnCCC	Undefined
EDIT	DTP Conversions	6	1/1/1998	12/31/2005	SxPnxxZnCCC	Undefined
EDIT	DTP Conversions	6	1/1/1998	12/31/2005	SxPnxxZZnCCC	Undefined

1. Select **ADD NEW** source.
2. Select a **Source Name** from drop down menu.
3. Input **Source Indicator** (first position of source key, i.e. 1=Renewals or E=Agent Conv)
4. Type in **Valid From** and **Valid To** dates (should select a date far in future, this field may be used when you change first position of source, i.e. if you change fulfillment houses). **AutoAdmin** looks at the select date on a key to determine if it fits the range.
5. **Create Setup Mask** - The Setup Mask identifies what part of the source key Admin will look at to create a campaign. Each fulfillment house has a different source key structure and therefore a different Setup Mask. To create a Setup Mask, choose from one of the following schemes below.

**NOTE:**

If you do not include a specific position for "P", panel, then the system will automatically default each key into a panel called "Control". Otherwise it will create a panel based on the panel scheme defined in the Panel Name Definition table, or create a panel based on the fulfillment code in the key, i.e. Panel A, Panel 1, etc.

**SAMPLE SETUP MASKS**

Since the **Setup Mask** is the most important component of a setup scheme, for reference here are some examples of setup masks by Fulfillment Company and source:

EDS/Centrobe	
Source	Setup Mask
Renewal (SIPE)	SxPxxZxCCC
Renewal (SPIE)	SPxxxZxCCC
Renewal (SIPE) - Double Digit	SxPxxZZxCCC
Renewal (SPIE) - Double Digit	SPxxxZZxCCC
Billing	SCCxCCxxxxZxxP
Billing - Double Digit	SCCxCCxxxxZZxxP
DTP	Depends on key structure
Gift	Depends on key structure
Palm Coast Data	
Source	Setup Mask
Renewal	SZCCPxCCCC
Billing	SZxCCCxPx
DTP	Depends on key structure
Gift	Depends on key structure
CDS	
Source	Setup Mask
Renewal	SCZxPPPxxxxCC (CC relates to effort select date not necessarily expire year)
Billing	SCZxPPPxxxxCC
DTP	Depends on key structure
Gift	Depends on key structure

## CAMPAIGN NAME DEFINITION

**Campaign Name Definitions** table allows you to define your campaigns based on the fulfillment codes and give them user-defined names. Naming your campaigns makes it easier to keep track of them and easier to do analysis. You can also define certain panel schemes for each campaign. **Example:** let's say you have an expire campaign with which you conducted a test, then you may want to create a panel scheme that will split the expire (two panels) – control and test. This table provides the ability to choose a panel scheme that automatically creates control and test panels.

**NOTE:**

If you already have prior campaigns setup in admin and want future source keys to be assigned to those campaigns, you still need to create a Campaign Name in this table. That campaign name must match the campaign and panel names exactly as they are already named in Admin for new keys to be correctly assigned to it. Otherwise, it will create a new campaign. If this happens, a quick fix is to delete the old or new campaign that was created, and reassign it correctly in the Campaign Name Definition table so that the keys can be reassigned. Deleting a campaign places those source keys back in the available keys table to be reassigned.

**Create Campaign Names:**

1. Select **ADD NEW** to create the campaigns for Bills and Renewals that you want the system to automatically add to Admin.
2. Type in a **Campaign Code** based on your "C" scheme in the setup mask (i.e. Centrobe Renewal CCC= 520, or Palm Coast CCCCCC = 1A2001).
3. Type in a **Campaign Description** name to identify your campaigns (i.e. "2000-01" for Renewal expires, or "99 BK 48" for Billing credit period).
4. Select a **Source Category** from the drop down menu.
5. Type in a **Campaign Date** (i.e. 1/1/2000 for a Jan expire) to assign.
6. Select a **Panel Scheme** from the drop down menu (based on what is defined in the Panel Name Definition table).
7. **Submit** to save

ADD NEW	Source Category	Campaign Code	Campaign Description	Campaign Date	Panel Scheme
EDIT	Billing	AX00	00 Credit Period AX00	8/1/2000	
EDIT	Billing	VI00	00 Credit Period VI00		
EDIT	Billing	0198	1998 BK 01	1/1/1998	Bill Default
EDIT	Billing	0298	1998 BK 02	1/1/1998	Bill Default
EDIT	Billing	0398	1998 BK 03	1/1/1998	Bill Default
EDIT	Billing	0498	1998 BK 04	1/1/1998	Bill Default
EDIT	Billing	0598	1998 BK 05	2/1/1998	Bill Default

**Panel Name Definitions** table is used to create your panel naming schemes. You can choose to identify all your panels as controls, or you can create a scheme for individual campaigns that may have tests going on and name them appropriately (i.e. control, price test, creative test, etc). The schemes you setup can then be applied to the **Campaign Name Definition** table.

**NOTE:**

If you already have prior campaigns setup in admin and want future source keys to be assigned to those campaigns, you still need to create a Campaign Name in this table. That campaign name must match the campaign and panel names exactly as they are already named in Admin for new keys to be correctly assigned to it. Otherwise, it will create a new campaign. If this happens, a quick fix is to delete the old or new campaign that was created, and reassign it correctly in the Campaign Name Definition table so that the keys can be reassigned. Deleting a campaign places those source keys back in the available keys table to be reassigned.

**Create Panel Name Definitions:**

1. Select **ADD NEW** to create a new panel scheme to define how campaign panels should be named and possibly split.
2. Type in a **Panel Scheme** name to define panel definition (i.e. Renewal Default, Jan 00 Test, Billing DM Test, etc).
3. Type in the **Package Code(s)** from the **Source Key (P)** that apply to the scheme (you can list more than one package at a time, just separate with a comma).
4. Type in the **Panel Name** you want to describe the panel for reporting (i.e. Control, Test 1, Price Test, Creative Test, etc.)
5. **Submit** to save

**Panel Name Definitions**

ADD NEW	Panel Scheme	Panel Codes	Panel Name
EDIT	Ack Retest	P,Z,U,D,G,J,S,T,Q,C,B,Z,1,2,3,9,*	Control
EDIT	Ack Retest	A	Control Panel
EDIT	Ack Retest	B	Test - Ack Retest
EDIT	Bill Default	A,B,C,D,E,F,G,H,J,K,L,M,N,P,Q,R,S,T,U,V,W,X,Y,Z	Control
EDIT	Bill Default	0,1,2,3,4,5,6,7,8,9,*	Control
EDIT	Creative Test	P,M,U,G,J,S,T,Q,B,C,Z,1,2,3,9,*	Control
EDIT	Creative Test	A,D	Control Panel
EDIT	Creative Test	B,E	Test - 1st Eff Ack Creative
EDIT	Email Test	P,M,A,D,G,J,S,T,Q,B,C,Z,1,2,3,9,*	Control
EDIT	Email Test	K	Control Panel
EDIT	Email Test	4	Test - Email
EDIT	Ren Default	A,B,D,E,G,J,K,Q,R,T,X	Control



## RENEWAL PRIOR SOURCE DEFINITIONS

**Using the Prior Source Definition Table**

In the case of some renewal sources, a user may wish to split keys into groups based on prior source. This can be accomplished by using the **Prior Source Definition** table. Common example might be taking a single conversion source and splitting it into two unique sources - DTP Conversions and Agent Conversions. The Prior Source Definition table provides a user the option of identifying which prior source codes are DTP versus Agent. If you have **Z** marked in your Setup Mask, then it will look to this table for more information on how to assign a key to the correct split source.

**NOTE:**

If prior source is identified in the source scheme setup mask ("Z"), but there are no corresponding records in the Prior Source Definition table, then the system will NOT be able to auto assign keys. Conversely, if there are records in the Prior Source Definition table, but prior source has not been identified in a source scheme setup mask ("Z"), then the system will NOT be able to properly assign keys.

**Create Prior Source Definitions:**

1. Select **ADD NEW** to create a **Prior Source Definition** table.
2. Select a **Source Name** from the drop down menu.
3. Input **Source Indicator** (1st position of source key, i.e. 1=Renewals or E=Agent Conv)
4. Input **Prior Source Indicator** (you can list more than one source at a time, just separate with a comma).
5. Input a **Valid From** and **Valid To** date that these prior source definitions are valid.
6. **Submit** to save

Renewal Prior Source Definitions					
ADD NEW	Source	Source Indicator	Prior Source Indicator	Valid From	Valid To
EXIT	Agent Conversions	6	A,B,C,D,E,F,G,H,I,J,K,L,M,N,P	1/1/1999	12/31/2005
EXIT	DTP Conversions	6	0,1,2,3,4,5,6,M,6,7,8,9,Q,R,S	1/1/1999	12/31/2005
EXIT	Renewals 1x	1	6	1/1/1999	12/31/2005
EXIT	Renewals 2+	1	1	1/1/1999	12/31/2005





Reporting on **N/S** (Newsstand Sale) Insert cards can be based on either **N/S Sales** or **N/S Draw**. IMT allows the user this flexibility through the **Mail Qty Override/Draw**, **Newsstand Sale**, and **Response based on Sale (Y)** or **Draw (N)** fields. These three fields, and two others, can all be found at the Key Levels and are described below:

**At the Key level, you will have the following fields to use:**

**Mail Qty** – This is the quantity that comes through from the fulfillment house and is a non-editable field. Take note; with Insert Cards, the fulfillment houses may not always key in this information on file. When the data that comes through this field is missing or in some cases not accurate, you will need to use the **Mail Qty Override/Draw** field.

**Mail Qty Override/Draw** – This is the field you will use to enter a **subscriber distribution** quantity or a **newsstand draw** quantity. Named **Mail Qty Override/Draw** because if you enter a number here, that number will override any number that comes through the fulfillment house in the **Mail Qty** field and the **expense set** will look to **calculate print costs** off this number. If there is no number input here, it will then look to the **Mail Qty** field to calculate expense.

**Newsstand Sale** – This is where you would enter a **quantity for sell-through** if you want to calculate your gross and net responses off this number, otherwise you can leave it blank.

**Response Based on Sale (Y), Draw (N)** – Enter a **Y** if you want the reports to use the **N/S Sale Qty** for response analysis, enter an **N** if you want the report to use the **Mail Qty Override Draw Qty** for response analysis. The **default** is set to **N**.

**Subtotal Flag** – This field is for reporting purposes. Generally, when you run a DTP report, you want to count all the mail quantities for each key in the total for that campaign (i.e. Direct Mail). However, with Insert Cards, the newsstand and subscription quantities for each card promoted should only be counted once in the campaign total, not once for each card. Enter an **F** or **False** if you do not want that key's mail quantity to be included in the subtotal. Enter a **T** or **True** if you do want that key's mail quantity to be included in the subtotal. The **default** is set to **T**.

## HOW TO START

Before running reports, you must organize your data using IMT's Admin application. When Admin is complete, you're ready to view reports. Here are the steps to either create a new report, or run a saved report:

### STEP 1 – LOGON TO THE SYSTEM

1. Go to <http://www.imtnetwork.com/>
2. Click "Client Center"
3. Login with your username and password
4. Click **Reporting**

### STEP 2 – CHOOSE NEW OR SAVED REPORT

#### Create a "NEW" report, follow these steps:

##### STEP 3 – SELECT A PRODUCT (S)

- Select One or More **Products**

##### STEP 4 – SELECT A REPORT

- Choose a **REPORT CATEGORY**
- Choose a **REPORT STYLE**

##### STEP 5 – NAME YOUR REPORT

- Naming your report is optional

**NOTE:** Name your report in order to **save** it (other than Undefined).

**Tip:** At this point you can run your report. However, if you want to filter your report data, then you will need to make a data selection.

##### STEP 6 – MAKE YOUR DATA SELECTION

Selection options include:

- Sources
- Campaign Type
- Campaign
- Panel Type
- Panel Test Type
- Panel Name

##### STEP 6 – RUN YOUR REPORT



- To run your report click the **RUN REPORT** button

##### STEP 7 – SAVE YOUR REPORT (optional)



- Go back to the report screen
- Click on the **SAVE** button
- Choose a **Report Group** (that has already been created) or **Create a New Report Group**
- Select a **Division / Group**
- **Go** and your report is now saved and stored by Division, Group and Report Group

#### Run a "SAVED" report, follow these steps:

##### STEP 3 – SELECT A DIVISION / GROUP

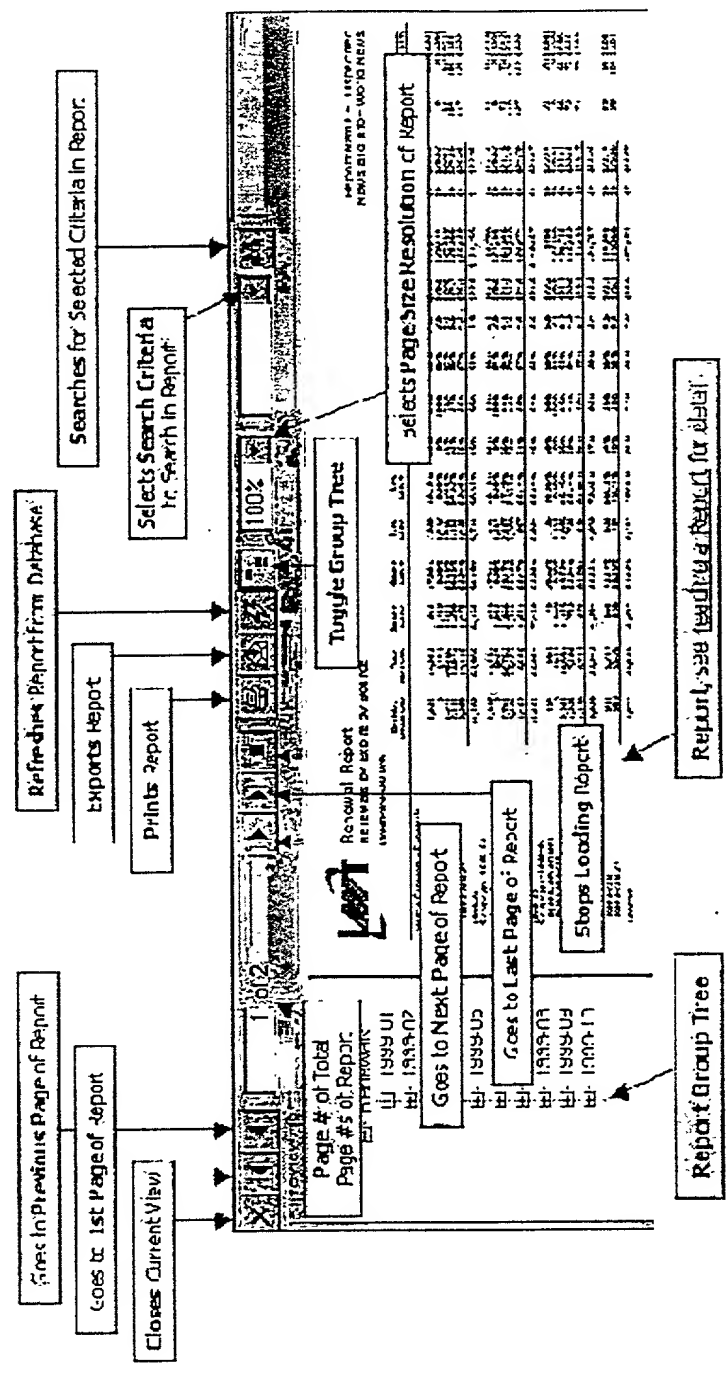
##### STEP 4 – SELECT A PRODUCT

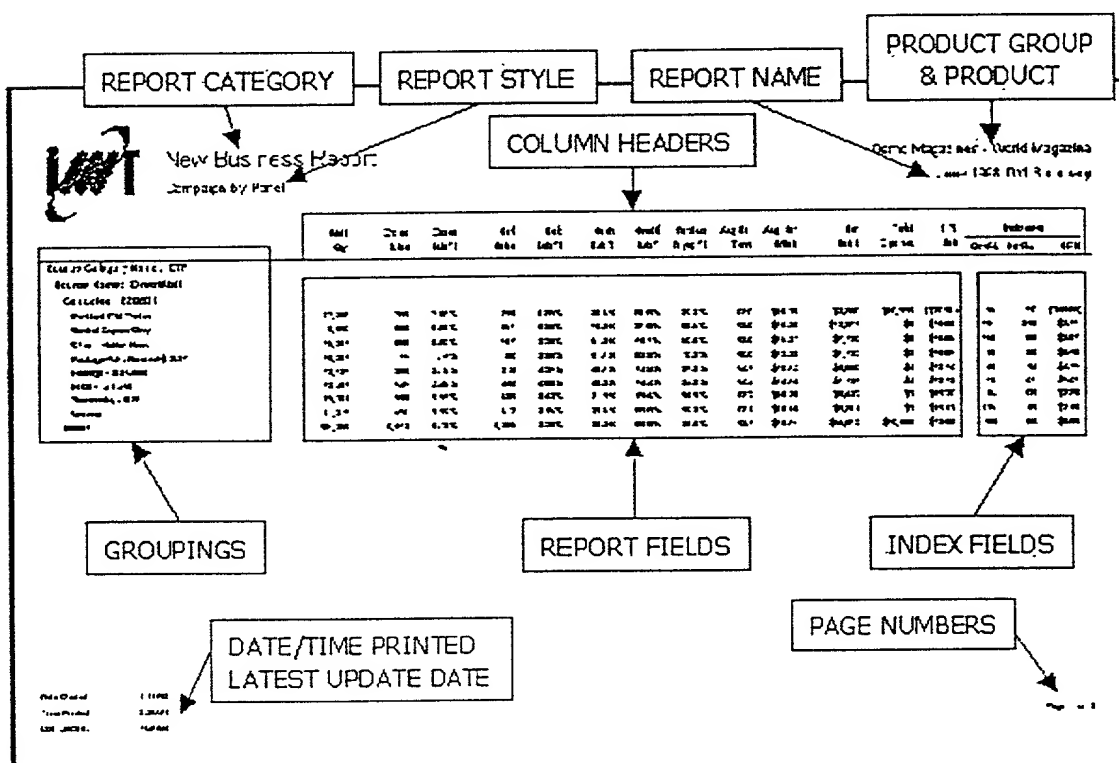
##### STEP 5 – SELECT A REPORT GROUP

- Click **RUN REPORT GROUP** to run all the reports within the specified group.
- Click **DELETE REPORT GROUP** to delete all the reports within a specified group
- Click **GO** to select a report within a report group

##### STEP 6 – SELECT A REPORT

- Click **RUN REPORT** to run an individual report
- Click **DELETE REPORT** to delete an individual report







IMT provides a variety of different **agent** reports for tracking agent production. As is the case with all IMT reports, these reports provide drill-down capabilities, which allow users to simply double-click on a field to view the detail below it. The standard fields on IMT **agent** reports include:

- Total Subs
- Gross \$
- Net \$
- Remit %
- Avg Term
- Avg Price
- Expenses
- NPN
- Rnwl %
- Cancels
- Reinstates
- Net Activity

For more information about field definitions, or how fields are calculated, please look to the table on the next page or refer to the glossary.



## FIELD DEFINITIONS

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
<b>Total Subs</b>	The total quantity of paid and credit subscriptions.	Summarized	The fulfillment house provides gross Sub quantities.
<b>Gross \$</b>	The dollar value of paid and credit subscriptions.	Summarized	The fulfillment house provides gross Revenue values.
<b>Net \$</b>	The dollar value of paid and credit subscriptions remitted to the publisher by the agent.	Summarized	The fulfillment house provides net Revenue values.
<b>Remit %</b>	Net \$ / Gross \$	Calculated	
<b>Avg Term</b>	Gross Copies / Total Subs	Calculated	Represents the average copies per subscription. The fulfillment house provides gross Copy quantities.
<b>Avg Price</b>	Gross \$ / Total Subs	Calculated	Represents the average dollar value per subscription.
<b>Expenses</b>	NA	NA	Expenses are NOT applied to agent reports at this time.
<b>NPN</b>	(Net \$ - Expenses) / Subs	Calculated	Represents the average net (remitted) value of subscriptions. Expenses are NOT included in NPN calculations at this time.
<b>Rnwl %</b>	Renewal Subs / Total Subs	Calculated	Renewal Subs for Agents are subscriptions that come in from an agent that match an existing record on file. The fulfillment house provides quantities.
<b>Cancels</b>	The total quantity of cancelled subs.	Summarized	The fulfillment house provides cancel quantities. Cancels correspond to cancellations received from an agent during a production period. Cancels do NOT correspond with Total Subs during the same period.
<b>Reinstates</b>	The total quantity of reinstated subs.	Summarized	Reinstate quantities are provided by the fulfillment house. Reinstates correspond to cancellations received from an agent during a production period. Reinstates do NOT correspond with Total Subs during the same period.
<b>Net Activity</b>	Total Subs - Cancels + Reinstates	Calculated	



## LIST OF AVAILABLE REPORTS

Below is a complete list of **agent** (Other reports are placed here) reports broken down by subcategory.

REPORTS BY SUB-CATEGORIES	SUB-CATEGORY DESCRIPTION
<b>SUMMARY ANALYSIS</b> <ul style="list-style-type: none"><li>• Publication by Source by Agent by Month</li><li>• Publication by Agent</li><li>• Publication by Source Month</li><li>• Publication by Source by Remit</li><li>• Publication by Source by Agent</li><li>• Agent by Publication</li><li>• Publication by Source</li><li>• <b>OTHER REPORTS</b><ul style="list-style-type: none"><li>• Starts by Model Source</li><li>• Starts by Source</li><li>• Available Keys</li><li>• Assigned Keys</li></ul></li></ul>	<p>Agent Summary reports offer the ability to monitor agent production by source, month and remit rate.</p>



IMT provides a variety of different **billing** reports for analyzing the performance of **billing** sources and campaigns. As is the case with all IMT reports, these reports provide drill-down capabilities, which allow users to simply double-click on a field to view the detail below it. Any campaign for which keys have been assigned is viewable through these reports. The standard fields on IMT **billing** reports include:

- Max Effort Qty
- Bills Mailed
- Total Payment
- Payment %
- Bills-Payment Ratio
- Step-Up Qty
- Step-Up %
- Request Cancel Qty
- Request Cancel %
- Total Revenue
- Total Expense
- NPN payment
- Payment On Payment
- Payment After Cancel
- Indexing – Pay-Up%, NPN

For more information about field definitions, or how fields are calculated, please look to the table on the next page or refer to the glossary.





FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
Max Effort Qty	Within a bill series the mail qty from the effort with the highest mailed quantity.	Calculated	Looks across all efforts to select the maximum billing pool size within a billing series.
Bills Mailed	The number of bills mailed for a particular effort.	Summarized	Bills Mailed quantities are generally provided by the fulfillment house.
Total Paymt	Represents the total number of payments.	Summarized	The fulfillment house generally provides payments.
Paymt %	Total Paymt / Bills Mailed or Total Paymt / Max Effort Qty	Calculated	The basis for calculating Paymt % will vary depending on the level and order of groupings. <ul style="list-style-type: none"><li>• For individual efforts Bills Mailed is used.</li><li>• For source, campaign, and panel subtotals, Max Effort Qty is used.</li></ul>
Bills - Paymt Ratio	Bills Mailed / Total Paymt.	Calculated	Represents the ratio of Bills Mailed to Total Payments.
Step-Up Qty	Step-ups are generally renewal orders made at the same time as a billing payment, usually in response to an offer on the billing effort.	Summarized	The fulfillment house generally provides step-up quantities.
Step-Up %	Step-Up Qty / Bills Mailed or Set-Up Qty / Max Effort Qty	Calculated	The basis for calculating Step-up % will vary depending on the level and order of groupings. <ul style="list-style-type: none"><li>• For individual efforts Bills Mailed is used.</li><li>• For source, campaign, and panel subtotals, Max Effort Qty is used.</li></ul>
Req Cnd	Request Cancels are cancels made by a subscriber.	Summarized	The fulfillment house generally provides Req Cnd quantities.
Req Cnd %	Req Cnd / Bills Mailed or Req Cnd / Max Effort Qty	Calculated	The method used to calculate Req Cnd % will vary depending on the level and order of groupings



## FIELD DEFINITIONS - CONTINUED

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
<b>Total Revenue</b>	Total payment dollars.	Summarized	The fulfillment house generally provides revenue quantities.
<b>Total Expenses</b>	Lettershop Costs+ Postage In + Postage Out + Premium Costs + Print Costs + Subs Service +Other Costs + Billing Costs + Bad Pay Costs + List Costs	Calculated	Expenses are dependent on (1) the association of an Expense Set to a campaign, panel or key, and (2) the values and settings that define an Expense Set.  Note: List Costs are the only expense item NOT reliant on an Expense Set. List Costs are applied directly to a key, and are calculated using either the mail qty or merge/purge qty as a basis.
<b>NPN Payment</b>	(Net Revenue - Total Expense) / Total Paymt	Calculated	Represents the average revenue per net payment.
<b>Pymt on Pymt</b>	A payment on payment is a second payment made on the same billing effort or series.	Summarized	The fulfillment house generally provides Payment On Payment quantities.
<b>Pymt After Cancel</b>	A payment after cancel is a payment made after a request cancel has been processed.	Summarized	The fulfillment house generally provides Payment After Cancel quantities.
<b>Indexes</b>			



## LIST OF AVAILABLE REPORTS

**Billing** reports are grouped into subcategories, ranging from high-level summary analysis to low-level effort analysis. Below is a complete list of **billing** reports broken down by subcategory.

REPORT SUB-CATEGORIES	DESCRIPTION
<b>SUMMARY ANALYSIS</b> <ul style="list-style-type: none"><li>Source by Current Source</li></ul>	Summary Analysis reports offer high-level views at the current source level. These reports are useful for comparing the overall performance of sources across credit periods.
<b>CREDIT PERIOD ANALYSIS</b> <ul style="list-style-type: none"><li>Source by Credit Period</li><li>Source by Current Source by Credit Period</li></ul>	Credit Period Analysis reports offer views at the credit period (campaign) level. These reports are useful for evaluating billing performance for a set of credit periods over a period of time.
<b>PANEL ANALYSIS</b> <ul style="list-style-type: none"><li>Current Source by Effort by Panel</li><li>Current Source by Panel</li></ul>	Panel Analysis reports offer views at the panel level. These reports are useful for comparing and evaluating panels within credit periods (campaigns), especially important when performing test panel analysis.
<b>EFFORT ANALYSIS</b> <ul style="list-style-type: none"><li>Panel by Effort</li><li>Source by Current Source by Effort</li><li>Source by Credit Period by Effort</li><li>Source by Effort by Current Source</li></ul>	Effort Analysis reports offer views at the effort level. These reports are useful for evaluating the performance of billing efforts for a group of credit periods or sources, or within credit periods over time.
<b>CHECKING REPORTS</b> <ul style="list-style-type: none"><li>Campaign Setup</li><li>Sourcekey Assignment</li></ul>	Effort Analysis reports offer views at the effort level. These reports are useful for evaluating the performance of billing efforts for a group of credit periods or sources, or within credit periods over time.



IMT provides a variety of different **DTP** (Direct to Publisher) Reports for analyzing the performance of **DTP** sources and campaigns. As is the case with all IMT reports, these reports provide drill-down capabilities, which allow users to simply double-click on a field to view the detail below it. Any campaign for which keys have been assigned is viewable through these reports. The standard fields on IMT **DTP** reports include:

- Mail Qty
- Gross Subs
- Gross Sub %
- Net Subs
- Net Sub %
- Cash Sub %
- Credit Sub %
- Total Sub Payup %
- Average Net Term
- Average Net \$/Sub
- Net Sub \$
- Total Expense
- NPN Sub
- Indexing – Grs%, Net %, NPN

For more information about field definitions, or how fields are calculated, please look to the table on the next page or refer to the glossary.



## FIELD DEFINITIONS

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
Mail Qty	Represents the quantity of promotions sent to a group of subscribers.	Summarized	The fulfillment house typically provides mail Qty. There is an option within Admin to override the fulfillment house mail quantity with an Override Mail quantity. Our formula will automatically use the Override Mail quantity if a value has been entered; otherwise we will use the mail quantity from the fulfillment house.
Gross Subs	Represents the number of paid and credit subscriptions sold.	Summarized	Gross subscription quantities are typically provided by the fulfillment.
Gross Sub %	Gross Subs / Total Mail Qty	Calculated	There are several ways users can influence the way Gross Sub % is calculated: <ol style="list-style-type: none"><li>1. If the user provides a Mail Qty Override quantity, then Mail Qty Override is used instead of the Mail Qty.</li><li>2. If "Newsstand Sale Qty" has been entered at the key level and response based on Sale "Y" Draw "N" is set to "Y" then the Newsstand Sale qty is used instead of substitute the Mail Qty.</li><li>3. In addition, there is an option to affect the way Gross Sub % is calculated for subtotal groupings. By default the system assumes ALL promotion keys will contribute to the total mail quantity, if the Subtotal Flag on a key is set to "FALSE" the mail qty for that key will not be involved in the total mail qty. Adjusting the subtotal flags is particularly useful for Insert Card reporting.</li></ol>
Net Subs	Represents the number of paid subscriptions sold (cash plus paid credit).	Summarized	Net subscription quantities are typically provided by the fulfillment.
Net Sub %	Net Subs / Total Mail Qty	Calculated	There are several ways users can influence the way Net Sub % is calculated -- see comments for Gross Sub %.



## FIELD DEFINITIONS - CONTINUED

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
Cash Sub %	Cash Subs / Gross Subs	Calculated	Represents the number of Cash Subscriptions as a percentage of Gross Subscriptions. Cash Subs are subscriptions received with money. Cash Sub quantities are generally provided by the fulfillment house.
Credit Sub %	Credit Subs / Gross Subs	Calculated	Represents the number of Credit Subs as a percentage of Gross Subs. Credit Sub quantities are generally provided by the fulfillment house.
Total Sub Payup %	Net Subs / Gross Subs	Calculated	Net Subs include both cash and paid credit.
Avg Net Term	Net Copies / Net Subs	Calculated	Represents the average number of copies per net subscription. The fulfillment house generally provides gross Sub Copy quantities.
Avg \$ / Net Sub	Net Revenue / Net Subs	Calculated	Net Revenue represents the dollar value for net subscriptions. Net Revenue is generally provided by fulfillment house.
Net Sub \$	Total Net Revenue	Summarized	Total Net Revenue is generally provided by the fulfillment house and represents the total dollar value of net subscriptions sold.
Total Expenses	Lettershop Costs + Postage In + Postage Out + Premium Costs + Print Costs + Subs Service + Other Costs + Billing Costs + Bad Pay Costs + List Costs	Calculated	Expenses are dependent on (1) the association of an Expense Set to a campaign, panel or key, and (2) the values and settings that define an Expense Set.  Note: List Costs are the only expense item NOT reliant on an Expense Set. List Costs are applied directly to a key, and are calculated using either the mail qty or merge/purge qty as a basis.
NPN Sub	(Net Revenue - Total Expense) / Net Subs	Calculated	Represents the average profit per net subscription sold.
Indexes			

# LIST OF AVAILABLE REPORTS

DTP reports are grouped into subcategories, ranging from high-level summary analysis to low-level list analysis. Below is a complete list of DTP reports broken down by subcategory.

REPORT CATEGORY	DESCRIPTION
<b>SUMMARY ANALYSIS</b> <ul style="list-style-type: none"> <li>Source Category by Source</li> </ul>	Summary Analysis reports offer high-level views at the source level. These reports are useful for comparing the overall performance of sources across campaigns.
<b>CAMPAIGN ANALYSIS</b> <ul style="list-style-type: none"> <li>Source by Campaign</li> <li>Source by Campaign Type by Campaign</li> </ul>	Campaign Analysis reports offer views at the campaign level. These reports are useful for evaluating campaign either individually or over a period of time.
<b>PANEL ANALYSIS</b> <ul style="list-style-type: none"> <li>Campaign by Panel Type by Test Type</li> <li>Campaign by Panel</li> <li>Campaign by Panel Subtype</li> <li>Campaign by Panel Subtype by Panel</li> <li>Campaign by Panel Test Type</li> <li>Campaign by Panel Test Type by Panel</li> <li>Campaign by Panel Type</li> <li>Campaign by Panel Type by Panel</li> <li>Panel Type by Panel</li> <li>Panel Type by Panel Subtype</li> </ul>	Panel Analysis reports offer views at the panel level. These reports are useful for comparing and evaluating panels within campaigns, especially important when performing test panel analysis.
<b>KEY ANALYSIS</b> <ul style="list-style-type: none"> <li>Campaign by Panel by Key</li> </ul>	Key Analysis reports offer views at the key level. These reports are useful for comparing and evaluating source keys and lists.
<b>LIST ANALYSIS</b> <ul style="list-style-type: none"> <li>List by List Segment</li> <li>List by List Segment by Campaign</li> <li>List Category by List</li> <li>List Category by List by List Segment</li> <li>List History (by List and Segment)</li> <li>Campaign by List</li> <li>Campaign by Panel by List</li> <li>Campaign by Panel by List and Segment</li> <li>Campaign by Panel by List by List Segment</li> <li>List by Campaign</li> </ul>	List Analysis reports offer views at the key level, with a specific focus on lists. These reports are particularly useful for evaluating the performance of list and list segments within a specific campaign, plus provide the ability to analyze the performance of list over time.
<b>PACKAGE ANALYSIS</b> <ul style="list-style-type: none"> <li>Source by Panel Type by Package</li> </ul>	Package Analysis enables comparison of creative within a source.
<b>CROSS TITLE ANALYSIS</b> <ul style="list-style-type: none"> <li>List by Title</li> <li>Source by Campaign Month by Title</li> <li>Source by Title</li> </ul>	Cross Title Analysis offer views at the source and campaign level. These reports are specifically designed to compare how titles are performing within the same sources.
<b>CHECKING REPORTS</b> <ul style="list-style-type: none"> <li>Campaign Setup</li> <li>Sourcekey Assignment</li> </ul>	Cross Title Analysis offer views at the source and campaign level. These reports are specifically designed to compare how titles are performing within the same sources.
<b>INTERNET</b> <ul style="list-style-type: none"> <li>Campaign by Panel by Key</li> <li>Campaign Type by Campaign</li> </ul>	Cross Title Analysis offer views at the source and campaign level. These reports are specifically designed to compare how titles are performing within the same sources.



IMT provides a variety of different **gift** reports for analyzing the performance of **gift** sources and campaigns. As is the case with all IMT reports, these reports provide drill-down capabilities, which allow users to simply double-click on a field to view the detail below it. Any campaign for which keys have been assigned is viewable through these reports. The standard fields on IMT **gift** reports include:

- Mail Qty
- Gross Orders
- Gross Orders%
- Gross Subs
- Gross Subs %
- Total Net Subs
- Net Sub %
- Total Payup %
- Cash Sub %
- Credit Sub %
- Avg Net Term
- Avg Net \$/Sub
- Sub-Ord Ratio
- Net Subs \$
- NPN Sub
- Indexing – GrOrds%, Gr Sub %, NPN

For more information about field definitions, or how fields are calculated, please look to the table on the next page or refer to the glossary.





## FIELD DEFINITIONS

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
Mail Qty	Represents the quantity of promotions sent to a group of subscribers.	Summarized	The fulfillment house typically provides mail Qty. There is an option within Admin to override the fulfillment house mail quantity with an Override Mail quantity. Our formula will automatically use the Override Mail quantity if a value has been entered; otherwise we will use the mail quantity from the fulfillment house.
Gross Orders	Represents the number of paid and credit subscription orders (donor responses). An order may include more than one subscription	Summarized	Gross Order quantities are generally provided by the fulfillment house
Gross Order %	Gross Orders / Total Mail Quantity	Calculated	There are several ways users can influence the way Gross Sub % is calculated -- see comments for Gross Sub %
Gross Subs	Represents the number of paid and credit subscriptions sold (donees and subscribing donors)	Summarized	Gross Sub quantities are generally provided by the fulfillment house
Gross Sub %	Gross Subs / Total Mail Qty	Calculated	<p>There are several ways users can influence the way Gross Sub % is calculated:</p> <ol style="list-style-type: none"><li>1. If the user provides a Mail Qty Override quantity, then Mail Qty Override is used instead of the Mail Qty.</li><li>2. In addition, there is an option to affect the way Gross Sub % is calculated for subtotal groupings. By default the system assumes ALL promotion keys will contribute to the total mail quantity, unless the Subtotal Flag on a key is set to "FALSE" the mail qty for that key will not be involved in the total mail qty. Adjusting the subtotal flags is particularly useful for Insert Card reporting.</li></ol>

**FIELD DEFINITIONS - CONTINUED**

<b>FIELD</b>	<b>DEFINITION / CALC METHOD</b>	<b>TYPE</b>	<b>COMMENT</b>
<b>Total Net Subs</b>	Represents the number of paid subscriptions sold (cash plus paid credit).	Summarized	Net subscription quantities are typically provided by the fulfillment.
<b>Net Sub %</b>	Net Subs / Total Mail Qty	Calculated	There are several ways users can influence the way Net Sub % is calculated -- see comments for Gross Sub %.
<b>Total Payup %</b>	Net Subs / Gross Subs	Calculated	Net Subs include both cash and paid credit.
<b>Cash Sub %</b>	Cash Subs / Gross Subs	Calculated	Represents the number of Cash Subscriptions as a percentage of Gross Subscriptions. Cash Subs are subscriptions received with money. Cash Sub quantities are generally provided by the fulfillment house.
<b>Credit Sub %</b>	Credit Subs / Gross Subs	Calculated	Represents the number of Credit Subs as a percentage of Gross Subs. Credit Sub quantities are generally provided by the fulfillment house.
<b>Avg Net Term</b>	Net Copies / Net Subs	Calculated	Represents the average number of copies per Net subscription. The fulfillment house generally provides gross Sub Copy quantities.
<b>Avg Net\$ / Sub</b>	Net Revenue / Net Subs	Calculated	Net Revenue represents the dollar value for Net subscriptions. Net Revenue is generally provided by fulfillment house.
<b>Sub-Ord Ratio</b>	Gross Sub / Gross Orders	Calculated	Represents the ratio of Gross Subs (donee+donor renewal order) to Gross Orders.
<b>Net Sub \$</b>	Total Net Revenue	Summarized	Total Net Revenue is generally provided by the fulfillment house and represents the total dollar value of net subscriptions sold.
<b>NPN Sub</b>	(Net Revenue - Total Expense) / Net Subs	Calculated	Represents the average profit per net subscription sold.
<b>Indexes</b>			



## LIST OF AVAILABLE REPORTS

Gift reports are grouped into subcategories, ranging from high-level summary analysis to low-level list analysis. Below is a complete list of **gift** reports broken down by subcategory.

REPORT CATEGORY	DESCRIPTION
<b>SUMMARY ANALYSIS</b> <ul style="list-style-type: none"><li>Source Category by Source</li></ul>	Summary Analysis reports offer high-level views at the source level. These reports are useful for comparing the overall performance of sources across campaigns.
<b>CAMPAIGN ANALYSIS</b> <ul style="list-style-type: none"><li>Campaign Type by Campaign</li><li>Source by Campaign</li></ul>	Campaign Analysis reports offer views at the campaign level. These reports are useful for evaluating campaign either individually or over a period of time.
<b>PANEL ANALYSIS</b> <ul style="list-style-type: none"><li>Campaign by Panel</li><li>Campaign by Panel Subtype</li><li>Campaign by Panel Subtype by Panel</li><li>Campaign by Panel Type</li><li>Campaign by Panel Type by Panel</li><li>Campaign by Test Type</li><li>Campaign by Test Type by Panel</li></ul>	Panel Analysis reports offer views at the panel level. These reports are useful for comparing and evaluating panels within campaigns, especially important when performing test panel analysis.
<b>EFFORT ANALYSIS</b> <ul style="list-style-type: none"><li>Source by Campaign by Effort</li><li>Source by Effort</li><li>Campaign by Panel by Effort</li><li>Source by Effort by Campaign</li></ul>	Effort Analysis reports offer views at the effort level. These reports are useful for evaluating the performance of efforts within a particular campaign, or for a group of campaigns.
<b>KEY ANALYSIS</b> <ul style="list-style-type: none"><li>Campaign by Panel by List</li><li>Campaign by Panel by Key</li></ul>	Key Analysis reports offer views at the key level. These reports are useful for comparing and evaluating source keys and lists.
<b>CROSS TITLE ANALYSIS</b> <ul style="list-style-type: none"><li>Source by Title</li></ul>	Cross Title Analysis offer views at the source and campaign level. These reports are specifically designed to compare how titles are performing within the same sources.
<b>CHECKING REPORTS</b> <ul style="list-style-type: none"><li>Campaign Setup</li><li>Sourcekey Assignment</li></ul>	



IMT provides a variety of different **renewal** reports for analyzing the performance of **renewal** sources and campaigns. As is the case with all IMT reports, these reports provide drill-down capabilities, which allow users to simply double-click on a field to view the detail below it. Any campaign for which keys have been assigned is viewable through these reports. The standard fields on IMT **renewal** reports include:

- Est. Max Effort Qty
- Total Mail Qty
- Gross Subs
- Gross Sub %
- Net Subs
- Net Sub %
- Cash Sub %
- Credit Sub %
- Total Sub Payup %
- Avg Term
- Avg \$/Sub
- Net Sub \$
- Expenses
- NPN sub
- Indexing – Grs %, Net %, NPN

For more information about field definitions, or how fields are calculated, please look to the table on the next page or refer to the glossary.



## FIELD DEFINITIONS

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
Est. Max Effort Qty	The largest mail quantity of a prior source group and renewal series within an expire pool.	Calculated	IMT automatically calculates the maximum effort by looking within an expire pool and finding the maximum mail quantity by prior source, regardless of the effort number. For example:  Expire 380  Prior Source A Effort 1 = 100 Effort 2 = 90 Max Effort = 100  Prior Source B Effort 1 = 90 Effort 2 = 100 Max Effort = 100 Total Max Effort = 200
Total Mail Qty	Represents the quantity of promotions sent to a group of subscribers.	Summarized	The fulfillment house typically provides mail Qty. There is an option within Admin to override the fulfillment house mail quantity with an Override Mail quantity, however this is generally not utilized for renewals.
Gross Subs	Represents the number of paid and credit subscriptions sold.	Summarized	Gross subscription quantities are typically provided by the fulfillment.
Gross Sub %	Gross Subs / Total Mail Qty or Gross Sub / Est. Max Effort Qty	Calculated	The basis for calculating Gross Sub % will vary depending on the level and order of groupings. <ul style="list-style-type: none"><li>• For individual efforts Total Mail Qty is used.</li><li>• For source, campaign, and panel subtotals, Est. Max Effort Qty is used.</li></ul>
Net Subs	Represents the number of paid subscriptions sold (cash plus paid credit).	Summarized	Net subscription quantities are typically provided by the fulfillment.



## FIELD DEFINITIONS - CONTINUED

FIELD	DEFINITION / CALC METHOD	TYPE	COMMENT
<b>Net Sub %</b>	Net Subs / Total Mail Qty or Net Sub / Est. Max Effort Qty	Calculated	The basis for calculating Net Sub % will vary depending on the level and order of groupings. <ul style="list-style-type: none"> <li>For individual efforts Total Mail Qty is used.</li> <li>For source, campaign, and panel subtotals, Est. Max Effort Qty is used.</li> </ul>
<b>Cash Sub %</b>	Cash Subs / Gross Subs	Calculated	Represents the number of Cash Subscriptions as a percentage of Gross Subscriptions. Cash Subs are subscriptions received with money. Cash Sub quantities are generally provided by the fulfillment house.
<b>Credit Sub %</b>	Credit Subs / Gross Subs	Calculated	Represents the number of Credit Subs as a percentage of Gross Subs. Credit Sub quantities are generally provided by the fulfillment house.
<b>Total Sub Payup %</b>	Net Subs / Gross Subs	Calculated	Net Subs include both cash and paid credit.
<b>Avg Term</b>	Gross Copies / Gross Subs	Calculated	Represents the average number of copies per gross subscription. The fulfillment house generally provides gross Sub Copy quantities.
<b>Avg \$ / Sub</b>	Gross Revenue / Gross Subs	Calculated	Gross Revenue represents the dollar value for gross subscriptions. Gross Revenue is generally provided by fulfillment house.
<b>Net Sub \$</b>	Total Net Revenue	Summarized	Total Net Revenue is generally provided by the fulfillment house and represents the total dollar value of net subscriptions sold.
<b>Expenses</b>	Lettershop Costs + Postage In + Postage Out + Premium Costs + Print Costs + Subs Service + Other Costs + Billing Costs + Bad Pay Costs + List Costs	Calculated	Expenses are dependent on (1) the association of an Expense Set to a campaign, panel or key, and (2) the values and settings that define an Expense Set.  Note: List Costs are the only expense item NOT reliant on an Expense Set. List Costs are applied directly to a key, and are calculated using either the mail qty or merge/purge qty as a basis.
<b>NPN Sub</b>	(Net Revenue - Total Expense) / Net Subs	Calculated	Represents the average profit per net subscription sold.
<b>Indexes</b>			



## LIST OF AVAILABLE REPORTS

Renewal reports are grouped into subcategories, ranging from high-level summary analysis to low-level effort analysis. Below is a complete list of **renewal** reports broken down by subcategory.

REPORT CATEGORY	DESCRIPTION
<b>SUMMARY ANALYSIS</b> <ul style="list-style-type: none"><li>• Source Category by Prior Source</li><li>• Source Category by Source</li><li>• Source Category by Source by Prior Source</li></ul>	Summary Analysis reports offer high-level views at the source level. These reports are useful for comparing the overall performance of source across expires (campaigns).
<b>EXPIRE ANALYSIS</b> <ul style="list-style-type: none"><li>• Expire by Source</li><li>• Source by Expire by Prior Source</li><li>• Source by Expire</li></ul>	Campaign Analysis reports offer views at the expire level. These reports are useful for evaluating expires (campaigns) either individually or over a period of time.
<b>PANEL ANALYSIS</b> <ul style="list-style-type: none"><li>• Source by Expire by Effort by Panel</li><li>• Source by Expire by Panel by Effort</li><li>• Source by Expire by Panel</li><li>• Source by Expire by Prior Source by Panel</li></ul>	Panel Analysis reports offer views at the panel level. These reports are useful for comparing and evaluating panels within expires (campaigns), especially important when performing test panel analysis.
<b>EFFORT ANALYSIS</b> <ul style="list-style-type: none"><li>• Prior Source by Effort</li><li>• Source by Effort</li><li>• Source by Effort by Prior Source</li><li>• Source by Expire by Effort</li><li>• Source by Prior Source by Effort</li></ul>	Effort Analysis reports offer views at the effort level. These reports are useful for evaluating the performance of efforts in an expire, or over time for a group of expires.
<b>TRENDING ANALYSIS</b> <ul style="list-style-type: none"><li>• Source by Effort by Expire</li><li>• Source by Prior Source by Expire</li></ul>	Trending Analysis reports offer views at the effort and prior source level. These reports are useful for evaluating the performance of efforts and prior sources over time for a group of expires.
<b>CROSS TITLE ANALYSIS</b> <ul style="list-style-type: none"><li>• Prior Source by Title</li><li>• Title by Prior Source</li><li>• Source by Title</li><li>• Source by Group</li></ul>	Cross Title Analysis offer views at the source and campaign level. These reports are specifically designed to compare how titles are performing within the same sources.
<b>CHECKING REPORTS</b> <ul style="list-style-type: none"><li>• Sourcekey Assignment</li><li>• Campaign Setup</li></ul>	

TERM	DEFINITION	EXAMPLES
<b>A</b>		
Available Keys	Set of active Source Keys that have not yet been assigned to a Panel	<ul style="list-style-type: none"> <li>See pages 8 &amp; 14</li> </ul>
Avg Net \$/Sub	Net Revenue ÷ Net Subscriptions	<ul style="list-style-type: none"> <li>\$12.00</li> </ul>
Avg Net Term	Net Copies ÷ Net Subscriptions	<ul style="list-style-type: none"> <li>12.0</li> </ul>
<b>B</b>		
Browser	Application that allows you to view information on the internet.	<ul style="list-style-type: none"> <li>Internet Explorer (IE)</li> </ul>
<b>C</b>		
Campaign	Marketing initiative consisting of one or more panels	<ul style="list-style-type: none"> <li>2000-01 Direct Mail</li> <li>2000-01 Insert Cards</li> </ul>
Campaign Status	Progress status of your campaign	<ul style="list-style-type: none"> <li>Open</li> <li>Closed</li> <li>Pending</li> </ul>
Campaign Type	Broad type classification of a Campaign which is user defined and can differ between sources	<ul style="list-style-type: none"> <li>Winter DM</li> <li>Hotline</li> </ul>
Company	Corporation, Inc., LLC, etc...	<ul style="list-style-type: none"> <li>ABC Direct Marketing Company</li> </ul>
Cash Sub %	Cash Subscriptions ÷ Gross Subscriptions	<ul style="list-style-type: none"> <li>40%</li> </ul>
Control Panel	Current offer / creative used as the standard against which to test any new offer / creative	<ul style="list-style-type: none"> <li>Control</li> </ul>
CPM	Cost per 1000 (M)	<ul style="list-style-type: none"> <li>\$75.00</li> </ul>
CPU	Cost per Unit	<ul style="list-style-type: none"> <li>\$1.50</li> </ul>
Credit Sub %	Credit Subscriptions ÷ Gross Subscriptions	<ul style="list-style-type: none"> <li>85.3%</li> </ul>



TERM	DEFINITION	EXAMPLES
<b>D</b>		
Drill Down	Data selection method in which you filter or 'drill' through your data by selecting specific criteria	<ul style="list-style-type: none"> <li>See page 8</li> <li>Run a report and double click on a Campaign Name and it will reveal the Panels within a Campaign, continue and you can drill down to the key</li> </ul>
Division	Group of Products within a Company	<ul style="list-style-type: none"> <li>Magazine Division</li> <li>Book Division</li> </ul>
<b>E</b>		
Edit View Screen	Screen in which you edit or delete records in Admin tables	<ul style="list-style-type: none"> <li>See pages 15 - 18</li> </ul>
Expense Set	Set of expenses which can be applied to a Campaign, Panel or Key	<ul style="list-style-type: none"> <li>2000-01 DM 6x9 Package Expense</li> <li>Triple Insert Expense</li> </ul>
Est. Max Mail Qty	Maximum quantity mailed between efforts within a Panel for all levels above effort	<ul style="list-style-type: none"> <li>Renewal Field Definitions – Page 49</li> </ul>
<b>F</b>		
Fast Find	Direct method to locate Sources, Campaigns and Panels within IMT Admin	<ul style="list-style-type: none"> <li>See page 8</li> </ul>
Filter Keys	Process of selecting specific Source Keys that you want to assign to a panel	<ul style="list-style-type: none"> <li>See page 14</li> </ul>
<b>G</b>		
Gross Subs	Total quantity of subs, cash + credit	<ul style="list-style-type: none"> <li>1000</li> </ul>
Group	Grouping of related products within a Division	<ul style="list-style-type: none"> <li>Arts Group</li> <li>Outdoor Group</li> </ul>
Group Edit	Feature that allows user to edit groups of records at one time	<ul style="list-style-type: none"> <li>See Pages 15 – 18</li> </ul>
Gross Sub %	Gross subscriptions ÷ Mail Qty	<ul style="list-style-type: none"> <li>60%</li> </ul>
<b>L</b>		
List	Source of list rental names	<ul style="list-style-type: none"> <li>Tattooing Weekly</li> <li>Money Monthly</li> </ul>
List Category	Broad category of related Lists	<ul style="list-style-type: none"> <li>House Lists</li> <li>Outside Lists</li> </ul>
List Segment	Specific selection from a List	<ul style="list-style-type: none"> <li>Paid Subscribers</li> <li>3 Month Buyers</li> </ul>
List Cost Factor	Provides the ability to calculate list expenses by factoring a percentage higher (or lower)	<ul style="list-style-type: none"> <li>If you have defined your actual list costs at \$100/M, but you want your reports to reflect a list cost of \$110/M, then would apply a factor 1.1</li> </ul>

TERM	DEFINITION	EXAMPLES
<b>M</b>		
Mail Qty	Quantity of Packages mailed for a specific Campaign, Panel, or Promotion Key	<ul style="list-style-type: none"> <li>1,000,000 Control Panel</li> <li>15,000 Test Panel 1</li> </ul>
Multi-Edit	Feature that allow user to edit one or more records at a time	<ul style="list-style-type: none"> <li>See Page 15 - 18</li> </ul>
<b>N</b>		
Net Sub \$	Net subscription revenue (cash + paid credit subscriptions) for a Campaign, Panel, or Promotion Key	<ul style="list-style-type: none"> <li>\$12,220</li> </ul>
Net Sub %	Net Subscriptions ÷ Mail Qty	<ul style="list-style-type: none"> <li>1.05%</li> </ul>
NPN Sub	Net per Net Subscribers	<ul style="list-style-type: none"> <li><math>(\text{Net Sub Rev} - \text{Total Exp}) \div \text{Net Subs} = \text{NPN Sub}</math></li> </ul>
NPN Pymt	Net Per Net Payment	<ul style="list-style-type: none"> <li><math>(\text{Total Rev} - \text{Total Exp}) \div \text{Total Pymt} = \text{NPN Pymt}</math></li> </ul>
<b>O</b>		
Optional Effort ID	Provides the ability to assign an effort sequence to keys. This is only valid for GIFT sources.	<ul style="list-style-type: none"> <li>Alphanumeric ID</li> </ul>
<b>P</b>		
Panel	Sub-grouping within a Campaign	<ul style="list-style-type: none"> <li>New Creative Panel</li> <li>Savings Certificate</li> </ul>
Panel Type	Set of related Panels typically used to distinguish Control Panels from Test Panels	<ul style="list-style-type: none"> <li>Control</li> <li>Test</li> <li>Newsstand</li> </ul>
Panel Subtype	Another field used to further define a set of related Panels	<ul style="list-style-type: none"> <li>Sweepstakes</li> <li>Bind-in</li> </ul>
Promotion Key - A.K.A - Source Key Data Key Document Key	Alphanumeric code that labels the element(s) to be measured in a Campaign. Source Keys are originated at fulfillment houses and standardized by IMT	<ul style="list-style-type: none"> <li>6KA2_0526</li> <li>41B2_08/21/2000</li> </ul>
Panel Test Type	Field used to define a set of related Panels by test method	<ul style="list-style-type: none"> <li>Price</li> <li>Offer</li> </ul>
Panel Package	Defines your package type	<ul style="list-style-type: none"> <li>6x9 green</li> </ul>
Product Group	Grouping of products within a Division	<ul style="list-style-type: none"> <li>Magazines</li> <li>Books</li> </ul>
Pymt on Pymt	Duplicate payment	<ul style="list-style-type: none"> <li></li> </ul>
Pay After Cancel	Payment received after subscription canceled	<ul style="list-style-type: none"> <li>Sub sends a payment when sub has been stopped</li> </ul>

TERM	DEFINITION	EXAMPLES
<b>R</b>		
Request Cnd	Gross Subscribers who have requested their Subscription to be canceled	<ul style="list-style-type: none"> <li>100</li> </ul>
Request Cnd %	$\text{Request Cnd} \div \text{Gross Subs}$	<ul style="list-style-type: none"> <li>30%</li> </ul>
Revenue Set	Revenue over and above Subscription revenue, applied to a Campaign, Panel or Key	<ul style="list-style-type: none"> <li>List Rental Revenue</li> <li>Advertising Revenue</li> </ul>

<b>S</b>		
Setup Table	<p>Tables which are used to define Campaigns, Panels or Keys in more detail (optional fields)</p> <p>See pages 9 - 10</p>	<ul style="list-style-type: none"> <li>AutoAdmin Setup</li> <li>Campaign Status</li> <li>Campaign Type</li> <li>Expense Set</li> <li>List Category</li> <li>List Name</li> <li>List Segment</li> <li>Panel Package</li> <li>Panel Sub Type</li> <li>Panel Test Type</li> <li>Panel Type</li> <li>Revenue Set</li> <li>Source Description</li> <li>Prior/Current Source Description</li> </ul>
Source	Specific channel of sale	<ul style="list-style-type: none"> <li>Direct Mail</li> <li>Renewals</li> </ul>
Source Category	Grouping of Sources (defined by IMT)	<ul style="list-style-type: none"> <li>Agents</li> <li>Billing</li> <li>DTP</li> <li>Gifts</li> <li>Renewals</li> </ul>
Sub Value	Dollar value of a sale of a Product or service for a specific time period	<ul style="list-style-type: none"> <li>\$14.95 / 1 Year</li> <li>\$24.00 / 2 Year</li> </ul>
Step Up Qty	Renewal at Birth or Duplicate Payment	<ul style="list-style-type: none"> <li>50 orders</li> </ul>
Step Up %	$\text{Step Up Quantity} \div \text{Bills Mailed}$	<ul style="list-style-type: none"> <li>5%</li> </ul>
Subscription	Sale of a Product or service for a specific time period	<ul style="list-style-type: none"> <li>Monthly paying customer/member</li> </ul>
Subtotal Flag	Used as a marker for calculating the mail quantity for subtotal. If the subtotal flag is set to False, then that key's mail quantity will NOT be included in subtotals	<ul style="list-style-type: none"> <li>Used monthly for Insert Cards</li> </ul>
Subtype	Same as Panel Type	<ul style="list-style-type: none"> <li>See Panel Type</li> </ul>
Sub - Ord Ratio	$\text{Subscriptions (donees)} / \text{Orders (donor) Ratio}$	<ul style="list-style-type: none"> <li><math>\text{Gross Sub} \div \text{Gross Orders} = \text{Sub-Ord Ratio}</math></li> </ul>

TERM	DEFINITION	EXAMPLES
<b>T</b>		
Term Value	Number of issues sold for a specific subscription sale	<ul style="list-style-type: none"> <li>• 12 Issues / 1 Year</li> <li>• 24 Issues / 2 Years</li> </ul>
Test Panel	New offer / creative tested against control	<ul style="list-style-type: none"> <li>• Smith 6x11, 14/\$35.00 12/23/99</li> </ul>
Total Rev	Total Revenue on Payments	<ul style="list-style-type: none"> <li>• \$2,200</li> </ul>
Total Exp	Total cost of a number of individual expense items	<ul style="list-style-type: none"> <li>• Lettershop + Printing + Postage Out + Postage In + Premium + Bad Pay + Billing + Subs Service + Expenses Other + List Expense = Total Exp</li> </ul>
Total Net Subs	Gross Subs that have paid, includes, Cash and Paid Credit Subscriptions	<ul style="list-style-type: none"> <li>• 50% cash</li> <li>• 50% paid credit</li> <li>• 75% = Total Net Subs</li> </ul>
Total Sub Pay up %	Net Subs ÷ Gross Subs	<ul style="list-style-type: none"> <li>• 50%</li> </ul>

Successful companies understand that knowing their customers, responding to their needs, and capitalizing on market share is the key to success. It's a belief that we share at IMT, because we know that "integrated marketing information is powerful". With a focus on the bottom line, our cutting-edge systems are used by some of the most progressive subscription-based companies today.

**IMT Reporting**

Provides highly segmented, clearly defined marketing analysis at almost any level – from source channel to campaign to list – regardless of the database size.

**IMT Modeling**

A sophisticated tool that businesses use to forecast trends and play what-if scenarios, with an emphasis on lifetime customer value.

**Consulting Services**

Services range from database design, business forecasting, retention program development, to strategic operations analysis.

- AGENT
- BILLING
- DTP
- GIFT
- RENEWAL